



UNIVERSITAT
POLITÈCNICA
DE VALÈNCIA

DSIC
DEPARTAMENT DE SISTEMES
INFORMÀTICS I COMPUTACIÓ

UNIVERSITAT POLITÈCNICA DE VALÈNCIA

Dpto. de Sistemas Informáticos y Computación

Diseño y desarrollo de un juego serio para aprender
pensamiento computacional aplicando gamificación

Trabajo Fin de Máster

Máster Universitario en Ingeniería y Tecnología de Sistemas
Software

AUTOR/A: Lara Mira, Antonio José

Tutor/a: Vos, Tanja Ernestina

Cotutor/a externo: MARIN CAMPUSANO, BEATRIZ MARIELA

CURSO ACADÉMICO: 2022/2023

Resumen

El software se ha convertido en una herramienta fundamental en la sociedad actual y está presente en prácticamente todos los aspectos de nuestra vida cotidiana para facilitar tareas que se realizan diariamente. El software aparece en sistemas de todo tipo, como pueden ser los industriales, de comunicación, entretenimiento, educación, transporte, medicina, y un largo etcétera ya que la lista puede ser casi interminable.

Aprender a programar se ha ido convirtiendo en una necesidad cada vez mayor, siendo la gamificación un mecanismo efectivo para el aprendizaje de la programación, ayudando a que las personas que quieren aprender estén más motivadas ante el reto de enfrentarse a entender como resolver problemas complejos a través del pensamiento computacional.

En la Universitat Politècnica de València (UPV), previamente se ha realizado un Trabajo Fin de Grado en el que se desarrolló un juego para iniciarse en la programación, enseñando el uso de algunas instrucciones que respetan la sintaxis del lenguaje de programación Python.

En este trabajo de Máster, se amplía este juego, agregando instrucciones de programación, nuevas funcionalidades y niveles de acuerdo al orden de contenidos plasmados en el programa de la asignatura de programación en la UPV. Además, en este trabajo se ha realizado una revisión del estado del arte a través de un mapeo sistemático para incluir los elementos más utilizados de gamificación en dicho juego. También se ha realizado una evaluación del aprendizaje a través de un experimento con estudiantes de ingeniería de la Universitat Politècnica de València durante el primer cuatrimestre del curso 22-23 que partían tanto con conocimientos previos en programación como sin ellos.

Los resultados obtenidos permiten concluir que al aplicar nuevos elementos de gamificación, como las recompensas tras superar niveles, una tabla de clasificación por los niveles superados, el uso de niveles y poder personalizar el robot, se consigue aumentar la motivación en el aprendizaje de la programación, ya que los estudiantes se divirtieron al utilizar el juego, considerándolo que es útil para iniciarse en el pensamiento computacional y en Python, abriendo un camino hacia la mejora de la educación.

Palabras clave: Gamificación, Python, programación, pensamiento computacional.

Resum

El software s'ha convertit en una eina fonamental en la societat actual i és present en pràcticament tots els aspectes de la nostra vida quotidiana per a facilitar tasques que es realitzen diàriament. El software apareix en sistemes de tota mena, com poden ser els industrials, de comunicació, entreteniment, educació, transport, medicina, i un llarg etcètera ja que la llista pot ser quasi interminable.

Aprender a programar s'ha anat convertint en una necessitat cada vegada major, sent la gamificació un mecanisme efectiu per a l'aprenentatge de la programació, ajudant al fet que les persones que volen aprendre estiguen més motivades davant el repte d'enfrontar-se a entendre com resoldre problemes complexos a través del pensament computacional.

A la Universitat Politècnica de València (UPV), prèviament s'ha fet un Treball Fi de Grau en el qual es va desenvolupar un joc per a iniciar-se en la programació, ensenyant l'ús d'algunes instruccions que respecten la sintaxi del llenguatge de programació Python.

En aquest treball de Màster, s'amplia aquest joc, agregant instruccions de programació, noves funcionalitats i nivells d'acord amb l'ordre de continguts plasmats en el programa de l'assignatura de programació en la UPV. A més, en aquest treball s'ha realitzat una revisió de l'estat de l'art a través d'un mapatge sistemàtic per a incloure els elements més utilitzats de gamificació en aquest joc. També s'ha realitzat una avaluació de l'aprenentatge a través d'un experiment amb estudiants d'enginyeria de la Universitat Politècnica de València durant el primer quadrimestre del curs 22-23 que partien tant amb coneixements previs en programació com sense ells.

Els resultats obtinguts permeten concloure que en aplicar nous elements de gamificació, com les recompenses després de superar nivells, una taula de classificació pels nivells superats, l'ús de nivells i poder personalitzar el robot, s'aconsegueix augmentar la motivació en l'aprenentatge de la programació, ja que els estudiants es van divertir al utilitzar el joc, considerant-lo com a útil per a iniciar-se en el pensament computacional i en Python, obrint un camí cap a la millora de l'educació.

Paraules clau: Gamificació, Python, programació, pensament computacional.

Abstract

Software has become a fundamental tool in today's society and is present in virtually all aspects of our daily lives to facilitate tasks performed on a daily basis. Software appears in systems of all kinds, such as industrial, communication, entertainment, education, transportation, medicine, and a long etcetera, as the list can be almost endless.

Learning to program has become a growing need, and gamification is an effective mechanism for learning programming, helping people who want to learn to be more motivated to face the challenge of understanding how to solve complex problems through computational thinking.

At the Universitat Politècnica de València (UPV), a Final Degree Project has been previously carried out in which a game was developed to get started in programming, teaching the use of some instructions that respect the syntax of the Python programming language.

In this Master's thesis, this game is extended, adding programming instructions, and new features and levels according to the order of content set out in the program of the programming subject at the UPV. In addition, in this work a review of the state of the art has been carried out through a systematic mapping to include the most used elements of gamification in this game. An evaluation of learning has also been carried out through an experiment with engineering students of the Universitat Politècnica de València during the first four-month period of the 22-23 course that started both with and without previous knowledge in programming.

The results obtained allow us to conclude that by applying new elements of gamification, such as rewards after overcoming levels, a ranking table according to the levels passed, the use of levels and the ability to customize the robot, it is possible to increase motivation in learning programming, since the students had fun using the game, considering it useful to get started in computational thinking and Python, opening a path towards improving education.

Keywords: Gamification, Python, programming, computational thinking.

Índice de figuras

1.1	Juegos para aprender instrucciones básicas de lenguajes de programación	6
1.2	Evolución de los lenguajes de programación según el índice TIOBE	7
1.3	Interfaz gráfica original del juego GIPPPY	8
1.4	Ejemplo de design science	9
2.1	Ejemplo de insignias	15
2.2	Framework MDA	15
2.3	Proceso de selección de artículos	18
2.4	Artículos con puntuación 1.	19
2.5	Artículos con puntuación 2.	19
2.6	Artículos con puntuación 3.	20
2.7	Artículos con puntuación 4 y 5.	20
2.8	Número de apariciones de los elementos en los artículos	23
2.9	Top-10 de elementos presentes en los artículos	24
3.1	Barra de idiomas	27
3.2	Juego en valenciano	28
3.3	Ejemplo de mapeo de idiomas	28
3.4	Barra de progreso y navegación	29
3.5	Insignias que puede conseguir el alumno	29
3.6	Barra de premios	29
3.7	Formulario de identificación del alumno	30
3.8	Formulario con la clasificación	30
3.9	Nivel 1	31
3.10	Premio de unas gafas al superar 1 nivel	32
3.11	Nivel 2	33
3.12	Premio de un gorro de fiesta al superar 3 niveles	34
3.13	Nivel 4 con obstáculo aleatorio	35
3.14	Insignia conseguida al superar 4 niveles	35
3.15	Nivel 6, consiguiendo como premio unas esposas	36
3.16	Clasificación tras superar 6 niveles	36
3.17	Nivel 7, incluyendo la brújula	37
3.18	Lazo como premio al superar 7 niveles	38
3.19	Nivel 8	38
3.20	Segunda insignia conseguida al superar 8 niveles	39
3.21	Nivel 9	39
3.22	Medalla como premio al superar 9 niveles	40
3.23	Nivel 10	41
3.24	Medalla como premio al superar 10 niveles	41
3.25	Nivel 11	42
3.26	Nivel 12	42
3.27	Coche al superar 12 niveles, y la tercera insignia	43
3.28	Nivel 13	43
3.29	Confeti final al superar todos los niveles	44

3.30	Clasificación tras superar 13 niveles	44
4.1	Arquitectura general de la aplicación	46
4.2	IDE para Python Pycharm	47
4.3	Herramienta utilizada para administrar bases de datos SQLite	48
4.4	Interfaz gráfica	49
5.1	IDE Visual Studio Code	51
5.2	Herramientas para desarrolladores del Chrome	52
5.3	Editor de gráficos Microsoft Paint 3D	52
5.4	Repositorio del código de la aplicación en GitHub	53
5.5	Estructura del código fuente	63
5.6	Estructura y código de la aplicación Ranking	64
6.1	Perfil de los participantes	72
6.2	Experiencia en programación	72
6.3	Niveles completados	73
6.4	Niveles más divertidos	73
6.5	Niveles más complicados	74
6.6	Preguntas UMUX	74
6.7	Preguntas UMUX	75
6.8	Elementos que más han gustado	75
7.1	Principales elementos incluidos en el juego	80
7.2	Análisis de accesibilidad del juego	81

Índice de tablas

1.1	Clasificación de los lenguajes de programación según el índice TIOBE	7
2.1	Elementos de diseño del juego	12
2.2	Categorías de elementos del juego	13
2.3	Dimensiones y elementos del juego enumerados en la literatura según [46]	14
2.4	Alternativas a la pregunta de investigación	17
2.5	Criterios de inclusión y exclusión.	17
2.6	Resultados de las búsquedas	21
2.7	Elementos presentes en los artículos	22

Índice general

Índice de figuras	1
Índice de tablas	2

Índice general	2
1 Introducción y motivación	5
1.1 Descripción de la solución	7
1.2 Objetivos	8
1.3 Resultados esperados	9
1.4 Metodología	9
1.5 Estructura del documento	10
2 Trabajos relacionados	11
2.1 Método de investigación	16
2.1.1 Pregunta de investigación	16
2.1.2 Estrategia de búsqueda	16
2.1.3 Extracción de información	17
2.2 Ejecución y análisis de resultados	21
2.3 Respuestas a la pregunta de investigación	23
3 Diseño del juego	27
3.1 Elementos comunes en todos los niveles	27
3.2 Nivel 1	30
3.3 Nivel 2	32
3.4 Nivel 3	33
3.5 Niveles 4 y 5	34
3.6 Nivel 6	35
3.7 Nivel 7	37
3.8 Nivel 8	38
3.9 Nivel 9	39
3.10 Nivel 10	40
3.11 Nivel 11	41
3.12 Nivel 12	42
3.13 Nivel 13	43
4 Diseño del software	45
4.1 Requisitos	45
4.2 Arquitectura	46
4.3 Interfaz de usuario	49
5 Implementación	51
5.1 Ampliaciones	53
5.2 Elementos de gamificación	58
5.3 Refactorización	65
6 Evaluación	71
6.1 Resultados	71
7 Conclusiones y trabajo futuro	79

CAPÍTULO 1

Introducción y motivación

Hoy en día el software se utiliza masivamente en diversos ámbitos de la vida cotidiana para facilitar tareas en sistemas de diversos tipos, por ejemplo en medicina, banca, educación, transporte, etc.

Aprender a programar puede ayudar a resolver los diferentes problemas que aparecen en el día a día de las personas que requieran del uso de tecnologías, y también a modo de fortalecimiento del desarrollo personal y profesional [51, 41].

Sin embargo, enseñar programación es un reto al que se enfrentan los educadores, ya que es difícil para los estudiantes el aprenderla al utilizar conceptos complejos y con un alto nivel de abstracción a los que no están acostumbrados. Esto puede provocar que los estudiantes se desmotiven, llevando a los educadores a buscar formas de enseñanza alternativas como puede ser a través de casos prácticos. El uso de juegos serios es una técnica que ayuda a motivar a quienes se quieren iniciar en la programación, como puede verse en diversas revisiones de la literatura [5, 16], siendo la gamificación una técnica considerada eficaz para aumentar la motivación de los estudiantes a la hora de enseñar programación [45], ayudando además al desarrollo del pensamiento computacional [25] necesario para la resolución de problemas.

El pensamiento computacional utiliza un conjunto de técnicas y habilidades propias de la computación para la formulación y resolución de problemas [3]. Está ampliamente reconocido por su importancia en informática y matemáticas, siendo útil también para todos los estudiantes, no solo para los científicos de la computación. Existen plataformas para la enseñanza de la codificación ^{1 2 3} que promueven el pensamiento computacional y las habilidades de resolución de problemas.

En este trabajo se utiliza la gamificación para ayudar en la enseñanza de la programación, ya que se ha demostrado que es una técnica eficaz en la educación, evidenciando que mejora el rendimiento de aprendizaje en estudiantes [32]. En educación, la gamificación es una estrategia que implementa los elementos de un juego en un contexto que no es del juego para resolver el problema con el objetivo de motivar al usuario en el proceso del aprendizaje. Al plantear retos a través de un juego se busca conseguir que de una forma entretenida el alumno dedique más tiempo al aprendizaje, haciéndole sentirse más atraído y comprometido.

Existen juegos que tratan de enseñar las instrucciones básicas de diferentes lenguajes de programación [55, 53, 54, 52] 1.1, los cuales están mayoritariamente creados para público en general, no para estudiantes universitarios; algunos requieren conocimientos

¹<https://scratch.mit.edu/>

²<https://code.org/>

³<https://csfirst.withgoogle.com/>

previos de lenguajes de programación y otros específicamente creados para el aprendizaje de programadores son de pago. Este es un problema con el que se encuentran los estudiantes al comenzar una carrera, ya que no tienen conocimiento previo en lenguajes de programación, con la desventaja adicional de encontrar juegos que no son gratuitos, por eso planteamos el uso del juego gratuito en la enseñanza universitaria.

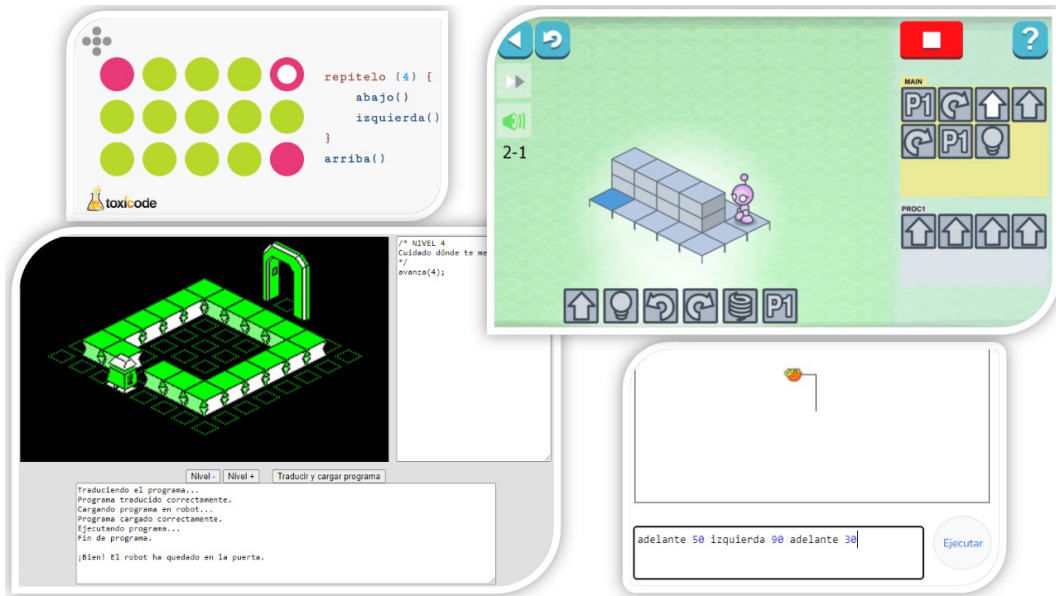


Figura 1.1: Juegos para aprender instrucciones básicas de lenguajes de programación

El juego utilizado en este trabajo se llama GIPPPY (Game for Introductory Programming Practice in Python), el cual está disponible en la web ⁴, y enseña los conceptos básicos del lenguaje Python. Se trata de un lenguaje de programación multiplataforma (puede ser utilizado en diversos sistemas operativos, como Windows, macOS y Linux), multiparadigma (admite diferentes estilos de programación, como programación orientada a objetos, programación funcional y programación estructurada) y de código abierto (su código fuente es público y está disponible para ser utilizado, modificado y distribuido por cualquier persona de forma gratuita). Es un lenguaje de alto nivel que se caracteriza por su sintaxis simple y expresiva, haciendo que esté próximo al lenguaje humano y sea sencillo de leer e interpretar. Python se ha usado masivamente para crear aplicaciones web, siendo popular para el aprendizaje automático y el análisis de datos ^{5 6}. Según el índice TIOBE⁷ en diciembre de 2022 python es el lenguaje en el que se han escrito la mayoría de las líneas de código 1.1, estando el lenguaje de programación C muy igualado. Viendo la evolución del uso de los lenguajes 1.2 parece confirmarse que Python es y será uno de los más utilizados.

⁴<https://robot.testar.org/>

⁵<https://www.python.org/about/>

⁶<https://www.simplilearn.com/python-features-article>

⁷<https://www.tiobe.com/tiobe-index/>

Dic 2022	Lenguaje	Calificaciones
1	Python	16,66 %
2	C	16,56 %
3	C++	11,94 %
4	Java	11,82 %
5	C#	4,92 %
6	Visual Basic	3,94 %
7	JavaScript	3,19 %
8	SQL	2,22 %
9	Ensamblador	1,87 %
10	PHP	1.62 %

Tabla 1.1: Clasificación de los lenguajes de programación según el índice TIOBE

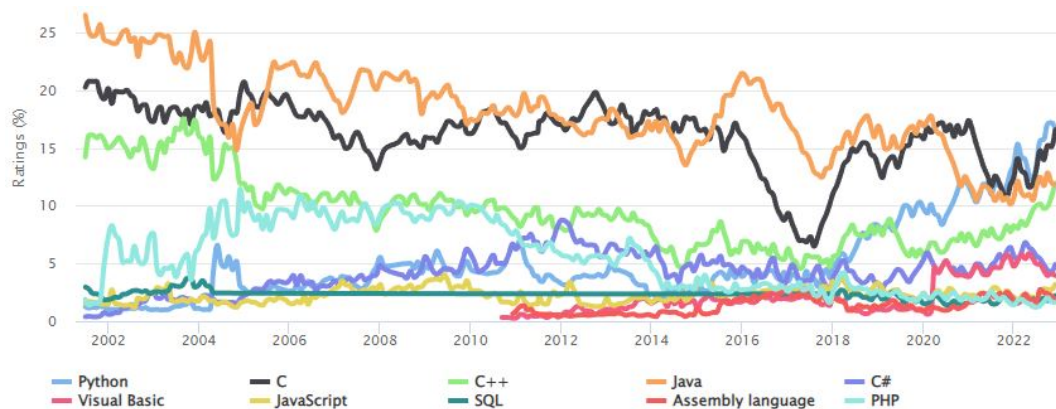


Figura 1.2: Evolución de los lenguajes de programación según el índice TIOBE

En la actualidad ya existen cursos de programación en la Universitat Politècnica de València que han pasado a utilizar el lenguaje de programación Python, con la consecuente adaptación del material didáctico, donde el uso del juego GIPPPY puede contribuir a iniciar a los alumnos de primer año en el aprendizaje de la programación a través de los retos que se plantean en los diferentes niveles, utilizando para ello elementos de gamificación que lo hagan más atractivo.

1.1 Descripción de la solución

Previamente, en el trabajo de fin de grado [47] se ha desarrollado una primera versión de un juego para iniciarse en la programación, el cual hace uso de la gamificación para que los estudiantes puedan practicar las habilidades de pensamiento computacional, aprendiendo instrucciones que respetan la sintaxis del lenguaje de programación Python, como puede verse en la figura 1.3. Las instrucciones del juego son AVANZA para mover el robot, GIRA para girarlo, y las estructuras SI como instrucción condicional utilizando condiciones fijas, y REPITE.



Figura 1.3: Interfaz gráfica original del juego GIPPPY

En el presente trabajo fin de máster se extiende este juego con más estructuras de programación, aportando además elementos de gamificación como recompensas y competitividad, y tratando de llegar a más usuarios utilizando nuevos idiomas. Para ello primeramente se ha realizado una revisión sistemática de los elementos de gamificación comúnmente utilizados en la literatura y en base a esto se ha desarrollado una nueva versión del juego incluyendo el top-10 de estos elementos.

Con el conocimiento de estos elementos de gamificación a ser incluidos, se ha diseñado y construido el juego de manera ágil, pudiendo contar con el juego funcional al terminar cada iteración. Actualmente el juego se encuentra en la web ⁸. La solución propuesta se ha desarrollado bajo la licencia open-source, permitiendo así que pueda seguir con su desarrollo y mejora.

1.2 Objetivos

El objetivo general es mejorar el desarrollo inicial de un juego para el aprendizaje de la programación y del pensamiento computacional mediante el uso de los elementos de gamificación más utilizados en la literatura, la relación de los niveles del juego con los contenidos enseñados en los cursos de programación, y la evaluación empírica del uso del juego serio por parte de los alumnos en los cursos de programación.

Al aportar más elementos de gamificación incluyendo nuevas funcionalidades que den más consistencia al juego se busca motivar a los estudiantes en su uso, ya que estos elementos hacen que dediquen más tiempo a jugar para superar los niveles, aparecer en la tabla de clasificación y personalizar el robot.

Para ello, se realiza una revisión del estado del arte a través de un mapeo sistemático, ya que será el encargado de hacer de guía hacia los elementos de gamificación a utilizar en el juego. La realización de un experimento aporta una validación práctica del aprendizaje obtenido.

⁸<https://robot.testar.org/>

1.3 Resultados esperados

Se espera que el uso del juego serio GIPPPY permita iniciar en el aprendizaje del lenguaje de programación Python de una forma más atractiva a las personas interesadas, viendo de forma aplicada conceptos utilizados en la programación como son las condiciones, los ciclos, los casos de prueba, u otros, y que también permita evaluar diferentes alternativas para la resolución de los retos planteados.

Por otro lado se espera que el juego sea utilizado en los primeros cursos de diferentes universidades, siendo la UPV, la universidad abierta de Holanda y la universidad Diego Portales de Chile las principales universidades donde será usado.

1.4 Metodología

Se ha investigado el estado del arte de la gamificación. La metodología utilizada en este trabajo es Design Science [11], la cual hace entender el objeto de estudio utilizando el diseño y la investigación como actividades principales.

Un punto importante es conocer el contexto y los objetivos que se quieren alcanzar en el trabajo, utilizando artefactos diseñados para interactuar con el contexto del problema de mejora. En particular, en este trabajo el artefacto es el juego, y el contexto el aprendizaje del pensamiento computacional, evaluando si se consigue mejorar el aprendizaje al hacer que dediquen esfuerzo y tiempo para resolver los problemas que el juego plantea de manera atractiva.

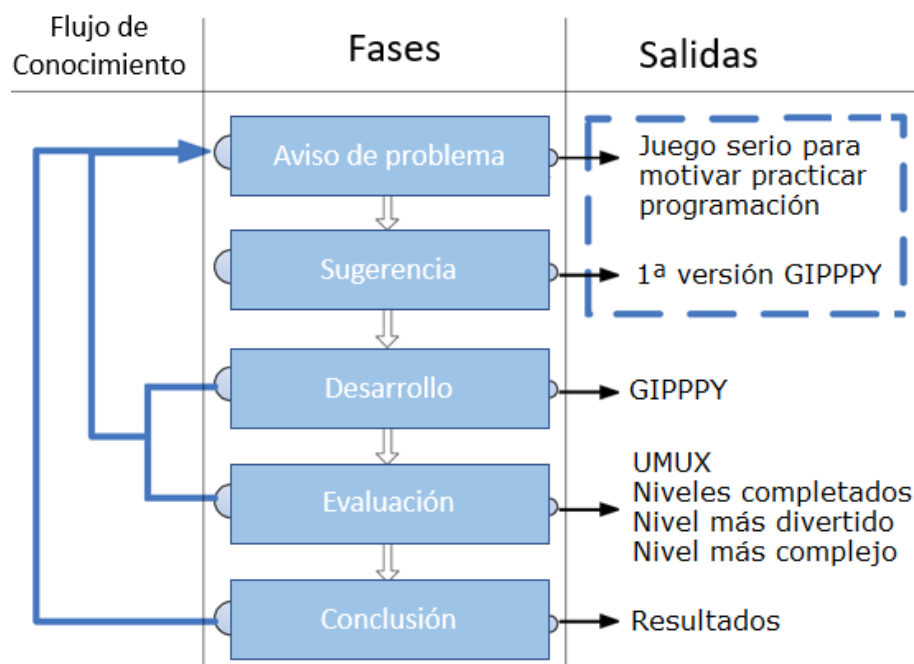


Figura 1.4: Ejemplo de design science

Tal como muestra la figura 1.4, la primera fase corresponde a la identificación del problema. Como se ha mencionado previamente, en los primeros cursos universitarios no se dispone de herramientas atractivas que faciliten a los estudiantes el aprendizaje del pensamiento computacional. Este hecho hace que se busque motivar a los estudiantes al uso de juegos serios que incluyan elementos de gamificación que les ayuden a practicar más para asimilar mejor los conceptos del temario. Es por esta razón que se ha diseñado y

desarrollado GIPPPY, un juego serio que ayude a practicar el pensamiento computacional en estudiantes de primer año de universidad. Para la evaluación del juego, se utilizará un cuestionario basado en UMUX [4].

Para el desarrollo del juego se ha seguido un enfoque de metodología ágil, donde se han realizado reuniones semanales con las tutoras donde se planteaban los requisitos, el método de verificación a utilizar para saber qué elementos de gamificación son más apropiados para fomentar el aprendizaje, revisando también el diseño y la implementación del juego, y se seleccionaba la siguiente característica a implementar o mejorar.

1.5 Estructura del documento

Este trabajo se estructura de la siguiente manera:

- En el capítulo 2 se revisa el estado del arte mediante un mapeo sistemático para encontrar los elementos de gamificación más utilizados.
- El capítulo 3 describe la interfaz gráfica del juego, presentando los niveles de juego, el objetivo de aprendizaje por nivel y las dinámicas que se han diseñado para lograr el objetivo de aprendizaje.
- El diseño del software y su arquitectura está expuesto en el capítulo 4.
- La implementación del juego se presenta en el capítulo 5, definiendo las principales funciones del juego.
- En el capítulo 6 se presenta el diseño y la ejecución de un experimento realizado para evaluar la percepción del juego por estudiantes de primer año de programación.
- El capítulo 7 finaliza el documento presentando las principales conclusiones y líneas de investigación futuras.

CAPÍTULO 2

Trabajos relacionados

A través de la gamificación y los juegos serios se quiere proporcionar a los estudiantes de herramientas que les ayuden a comprender el pensamiento computacional y les motiven practicarlos.

En el artículo [9] se indica que fue en 1970 cuando se utilizó por primera vez el término "juego serio" por Clark Abt en su libro *Serious Games* [1], aplicándolo a videojuegos diseñados con fines educativos. Hoy en día se usa en un contexto más amplio, considerándose juego serio cuando la finalidad del juego va más allá del entretenimiento. Los juegos serios y la gamificación son tratados como enfoques utilizados en contextos educativos para estimular y motivar a los estudiantes con el fin de obtener un mayor rendimiento en su aprendizaje, utilizando para ello elementos del juego, como puedan ser un sistema de puntos, insignias o tablas de clasificación.

En [23] se presenta un catálogo de requisitos de gamificación, los cuales fueron identificados tras realizar una revisión sistemática de la literatura. Según los autores de [23], la gamificación es un enfoque serio para acelerar la curva de experiencia del aprendizaje, e indican que la tarea de identificar qué elementos del juego se deben aplicar es problemática y no se ha llegado a un consenso. También hace referencia a una afirmación de Landers sobre las muchas maneras en las que puede ser usada la gamificación en la educación, por ejemplo las puntuaciones ayudan a ver la evolución de un alumno, y el desafío podría usarse en una actividad de discusión en pequeños grupos para competir por la "mejor" respuesta. Por otro lado se indica que la presión del tiempo puede considerarse un aspecto crítico al agregar tensión, lo cual puede provocar que se dificulte el aprendizaje, por lo que se deben priorizar los elementos a utilizar para saber cuáles son los más recomendables para cada tipo de estudiante. Este sería un ejemplo diferenciador entre la gamificación aplicada a los juegos serios y a los juegos de lúdicos, donde el límite de tiempo es utilizado habitualmente en juegos de entretenimiento sin tener en cuenta si se trata de un método eficaz al establecer objetivos claros que estimulen el aprendizaje.

En la literatura se menciona el uso de elementos de juego, pero no se suelen mencionar los entornos de trabajo de gamificación que integren los requisitos para el aprendizaje de lenguajes de programación. En [20] se propone un framework de gamificación que analiza la dificultad de enseñar y aprender java. Clasifica los elementos de diseño del juego, diferenciando entre elementos de mecánica, los cuales son parte de las actividades o elementos diseñados para ser usados en el juego, y los elementos de dinámica, que corresponden a la interacción que el jugador tiene con el juego, incluyendo también la motivación que está relacionada con el tipo de motivación que se logra cuando se utiliza la mecánica según la dinámica diseñada. En su trabajo se habla de integrar los requisitos de aprendizaje y los elementos del juego para superar los problemas de aprendizaje. La tabla 2.1 muestra los elementos de juego que abordan.

Elementos de diseño del juego		Motivos
Elementos del juego: Mecánica	Elementos del juego: Dinámica	
Documentación del comportamiento	Exploración	Curiosidad intelectual
Sistemas de puntuación, insignias, trofeos	Recopilación	Logro
Clasificaciones	Competición	Reconocimiento social
Rangos, niveles, puntos de reputación	Adquisición de estatus	Reconocimiento social
Tareas de grupo	Colaboración	Intercambio social
Presión de tiempo, tareas, misiones	Desafío	Estimulación cognitiva
Avatares, mundos virtuales, comercio virtual	Desarrollo / Organización	Autodeterminación

Tabla 2.1: Elementos de diseño del juego

Se ha encontrado un artículo [21] donde se explica cómo gamificar el aprendizaje de una manera científicamente respaldada proporcionando ejemplos aplicados. Landers (2014) define la gamificación del aprendizaje como “el uso de elementos del juego, incluido el lenguaje de acción, la evaluación, el conflicto/desafío, el control, el entorno, la ficción del juego, la interacción humana, la inmersión y las reglas/objetivos, para facilitar el aprendizaje y los resultados relacionados”. La tabla 2.2 muestra la lista de estos elementos.

Atributo	Teoría	Definición
Lenguaje de acción	Teoría de la presencia	El método y la interfaz mediante los cuales se produce la comunicación entre un jugador y el juego en sí
Evaluación	El efecto de las pruebas	El método por el cual se rastrean los logros y el progreso del juego
Conflicto/desafío	Teoría del establecimiento de objetivos	Los problemas que enfrentan los jugadores, incluida la naturaleza y la dificultad de esos problemas
Control	Teoría de la auto-determinación	El grado en que los jugadores pueden alterar el juego y el grado en que el juego se altera a sí mismo en respuesta
Ambiente	Teoría de la presencia	La representación del entorno físico del jugador
Ficción del juego	La hipótesis narrativa	El mundo del juego ficticio y la historia
Interacción humana	Constructivismo social	El grado en que los jugadores interactúan con otros jugadores tanto en el espacio como en el tiempo
Inmersión	Teoría de la presencia	La experiencia afectiva y perceptual de un juego
Reglas/objetivos	Teoría del establecimiento de objetivos	Reglas, objetivos e información claramente definidos sobre el progreso hacia esos objetivos, proporcionados al jugador

Tabla 2.2: Categorías de elementos del juego

En [6] recomiendan que las aplicaciones no contengan sanciones (sonidos de desaprobación, aspa roja) ni penalizaciones al ser percibidos como desalentadores, mientras que las recompensas las valoran positivamente. Proponen que en lugar de penalizar a los usuarios por acciones incorrectas, que el software educativo brinde asistencia en el juego en forma de ayuda correctiva o retroalimentación.

Se han revisado investigaciones que analizan si vale la pena utilizar la gamificación en la educación, como por ejemplo en [36] donde se realiza un estudio experimental que evalúa el uso de la gamificación en la educación de pruebas de software, utilizando una plataforma digital que incluía 10 elementos de juego para aumentar la motivación y rendimiento de los estudiantes, donde, a pesar de que el desempeño no mejoró, concluyen que vale la pena gamificar al aportar beneficios como la atención de los estudiantes, la motivación, participación, y clases mas amenas y divertidas.

Hay trabajos que tratan de evidenciar de manera empírica los efectos que la aplicación de elementos del juego puede generar en los usuarios. En [18] realizan un estudio donde evalúan 21 patrones de diseño de juegos en relación con la mejora que puede generar en el rendimiento de aprendizaje, logro de objetivos y compromiso en los estudiantes al utilizar un curso masivo online gratuito (MOOC) en un taller de diseño de juegos, identificando los siguientes elementos como candidatos a obtener los efectos deseados: empoderamiento, curvas de aprendizaje suave, canales de comunicación, niveles, pistas, indicadores de metas, árboles de habilidades, premios y narrativa. En [43] realizan un estudio experimental aplicado a 22 alumnos universitarios para enseñar el proceso de medición de software utilizando elementos de juegos serios para fomentar el interés y

la interacción de los estudiantes creando un ambiente competitivo y cooperativo, viendo que la gamificación tiene un efecto positivo cuando se usa de forma inteligente al centrarse en los aspectos sociales, cognitivos y emocionales.

El uso de modelos puede ser utilizado por los desarrolladores de juegos como una herramienta de rastreo de elementos específicos para conseguir una mejora en el rendimiento del usuario. En [46] presentan un modelo unificado de gamificación y motivación (UGM) con las relaciones entre elementos del juego, fuentes de motivación, y la relación entre el compromiso del usuario y el desempeño. Revisan estudios relacionados con la gamificación, enumerando la cantidad de veces que una dimensión o un elemento es nombrado, como puede verse en la tabla 2.3.

Dimensión									
Propósito	11	Retro-alimentación	98	Propiedad	16	Reto	6	Premio	4
Elemento									
Metas	2	Puntos	24	Elección	2	Salud	1	Moneda virtual	1
Narrativa	3	Insignias	27	Avatar	7	Tiempo límite	1	Premio	1
Héroe	1	Clasificación	24	Personalizar	1	Competición	4	Recompensa	2
Suerte	2	Niveles	8	Salas chat	1				
Otros	3	Barra de progreso	4	Votación	2				
		Desbloques	1	Marcos	3				
		Otros	10						

Tabla 2.3: Dimensiones y elementos del juego enumerados en la literatura según [46]

Se han encontrado trabajos donde se presentan juegos educativos aplicados a la programación, como en [19] con Reduct, un juego que enseña conceptos básicos de programación expresados en JavaScript. En este trabajo remarcan una preocupación en el uso de elementos de recompensa (insignias, puntos o narrativa) en los juegos educativos ya que pueden distraer la atención durante el aprendizaje.

En [29] se habla de *gamificación básica* cuando se añaden algunos elementos de gamificación a sistemas previamente diseñados, buscando una mejora en la motivación extrínseca, siendo habitual el uso de puntos, insignias y/o tablas de clasificación. La *gamificación compleja* sería un rediseño del sistema, utilizando mecánicas de juego más elaboradas, requiriendo de un mayor esfuerzo que permitirá a su vez mejorar la motivación intrínseca, por ejemplo utilizando herramientas o plataformas software de gamificación preexistentes para mejorar el proceso de aprendizaje.

Las insignias son elementos muy utilizados para mejorar los sistemas educativos 2.1, los cuales afectan a los estudiantes de manera positiva o negativa en su comportamiento. En [10] realizan un estudio para comprender cómo pueden afectar las insignias a los estudiantes, sugiriendo que no entorpece a la mayoría de los estudiantes al no afectar negativamente al aprendizaje, siempre que la gamificación no se vuelva más importante que la función real del juego, en cuyo caso los estudiantes se centrarían en obtener más insignias o puntuación en lugar de adquirir conocimiento, incluso al lograr insignias en competición con el tiempo puede tener un efecto negativo.



Figura 2.1: Ejemplo de insignias

En [40] se realiza una revisión sistemática de la literatura respecto a las estrategias de gamificación en el proceso de aprendizaje utilizando herramientas de aprendizaje como el juego de la serpiente, viendo que puede ser aplicado como parte del concepto de la estrategia de gamificación en la educación infantil, aumentando el atractivo, la interacción y la comprensión. En niveles de educación superiores como primaria o secundaria se implementan con éxito las estrategias de gamificación para el uso de multimedia interactiva, el aprendizaje de la animación y del sistema contable, al contener unos patrones de pensamiento más fáciles de dirigir en edades más avanzadas.

A través del artículo [33] se realiza una investigación cuyo propósito es comprender cómo los elementos de gamificación propuestos que utilizan el Framework Mecánica, Dinámica y Estética (MDA: Mechanics-Dynamics-Aesthetics) mejoran la motivación de los usuarios y los resultados de aprendizaje, concluyendo que la mayoría de los estudiantes consiguen estar más motivados para aprender inglés al utilizar los elementos de gamificación propuestos. También se ha revisado el artículo [44], donde se analiza el uso de la gamificación en el aprendizaje utilizando el Framework MDA, revisando la literatura para detallar el modelo utilizado.

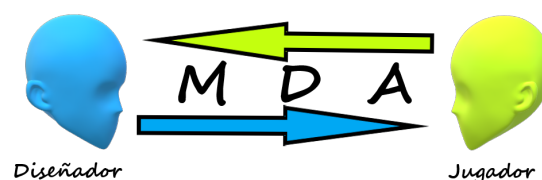


Figura 2.2: Framework MDA

La investigación realizada en [39] hace hincapié en la importancia del sistema de recompensas en la gamificación al motivar a los estudiantes a realizar las actividades planteadas en el curso, viendo como el grupo de estudiantes con elementos de recompensa se sienten más motivados que el grupo que no tenía el sistema de recompensas, sino puntuaciones. Por otro lado, en [15] argumenta que la gamificación aplicada hoy en día es limitada al enfocarse en elementos de recompensas, alentando a los investigadores a revisar como incorporar los elementos en sus trabajos.

Al analizar estos trabajos aparecen diferentes elementos que pueden aplicarse al juego para motivar a los alumnos en el aprendizaje de los lenguajes de programación y del pensamiento computacional. Para identificar claramente cuáles son los más utilizados o que aportan mayores beneficios se ha realizado un mapeo sistemático, el cual se presenta a continuación.

2.1 Método de investigación

Para analizar el estado del arte se realiza un mapeo sistemático de la literatura, profundizando en los elementos de gamificación utilizados para mejorar el aprendizaje al utilizar juegos serios en la docencia, revisando las diferentes técnicas propuestas para resolverla. Este mapeo consiste en un estudio no sesgado que lleva a identificar trabajos relevantes relacionados con el ámbito de este trabajo, identificando una idea general entre la información obtenida.

Se ha evaluado esta integración y su aportación a mejorar la docencia, tratando de encontrar evidencias de estudios sobre juegos serios y elementos de gamificación aplicados a la enseñanza de la programación para tratar de identificar qué técnicas se pueden aplicar al desarrollo del juego del presente trabajo.

Se ha seguido el proceso definido por Kitchenham [7], que indica que se debe definir una pregunta de investigación con la finalidad de revisar la literatura en busca de la identificación de la información más relevante que lleva a la realización del mapeo.

2.1.1. Pregunta de investigación

La pregunta de investigación es la que indica el objetivo de la revisión, y para este trabajo consiste en ver qué elementos de gamificación ayudan a entender el pensamiento computacional, lo cual lleva a utilizar la siguiente pregunta:

RQ: What elements of gamification are used in serious games for programming?

DATA ACTION: Lenguajes y cuales demuestran una mejora en el aprendizaje.

La revisión está estructurada por los siguientes criterios PICOC [2], por su sigla en inglés:

- **Población:** Estudios sobre juegos serios aplicados en la docencia.
- **Intervención:** Juegos que utilizan elementos de gamificación.
- **Comparación:** Los trabajos son comparados según los elementos utilizados y su aportación.
- **Resultado:** Identificar los elementos de gamificación más utilizados.
- **Contexto:** Se buscan evidencias en toda la literatura, sin definir restricciones de contexto.

2.1.2. Estrategia de búsqueda

Para el proceso de búsqueda se han utilizado diferentes bibliotecas digitales, lo que ha permitido realizar búsquedas más óptimas y que eviten fallos en el proceso.

Los buscadores de las bibliotecas digitales permiten el uso de palabras clave, por lo que se han extraído de la pregunta de investigación las palabras: juegos, programación, elementos y gamificación.

El proceso de búsqueda se realizó en cuatro bibliotecas digitales: ACM Digital Library, IEEE Xplore, Scopus Preview y SpringerLink. Estas bibliotecas fueron seleccionadas por ser utilizadas para publicar los trabajos de las revistas y conferencias más relevantes en el área de estudio.

Al comenzar a aplicar la búsqueda en las diferentes bibliotecas con la pregunta de investigación inicial no se encontraron artículos para la biblioteca Scopus, y tan solo 16 para IEE Xplore, mientras que para SpringerLink se encontraron 1977 artículos. Ante esta situación se optó por modificar la pregunta de investigación para realizar una nueva fase de búsqueda utilizando preguntas alternativas, por lo que finalmente utilizamos las siguientes 3 preguntas:

- *Pregunta 1: What elements of gamification are used in serious games for programming.*
- *Pregunta 2: Elements of gamification used in serious games for programming.*
- *Pregunta 3: Elements of gamification used in serious games.*

Se utilizan las preguntas en inglés debido a que en las Ciencias de la Computación la mayoría de las publicaciones se realizan en inglés.

La tabla 2.4 resume los resultados obtenidos en las bibliotecas digitales utilizadas para cada una de estas preguntas, apareciendo remarcado en negrita la opción de resultados que ha sido utilizada en la investigación. Para la biblioteca SpringerLink se aplicó además una restricción adicional donde se indica la subdisciplina *User Interface & HCI*, ya que los elementos de gamificación se encuentran en la interfaz gráfica de usuario y en la interacción con el sistema, consiguiendo así reducir el número de artículos a revisar.

		ACM Library	IEE Xplore	Scopus Preview	Springer Link			
Filtro					Todo	Computer Science	Computers & Education	UI & HCI
Pregunta	1	133	16	0	1977	719	214	149
	2	604999	16	9	2190	816	248	172
	3	591588	100	235	2828	1165	381	285

Tabla 2.4: Alternativas a la pregunta de investigación

2.1.3. Extracción de información

Se ha accedido a las diferentes bibliotecas digitales, utilizando en sus buscadores nuestra cadena de búsqueda. Una vez obtenidos los resultados se procede a aplicar los siguientes criterios de inclusión y exclusión:

Criterios de inclusión	Criterios de exclusión
- Artículos en inglés y en español	- Artículos que no sean científicos o conferencias
- Artículos relacionados con elementos de gamificación	- Artículos ya seleccionados en revisiones anteriores

Tabla 2.5: Criterios de inclusión y exclusión.

Seguidamente se procede a leer el título y el resumen de los artículos candidatos, definiendo en cada uno si es útil o no. Los que son útiles se continúa con la lectura de la introducción, por lo que ya se podría decidir si se incluye a la lista de aceptados, y en caso de tener dudas se lee el artículo completo. La figura 2.3 muestra el proceso completo de selección.

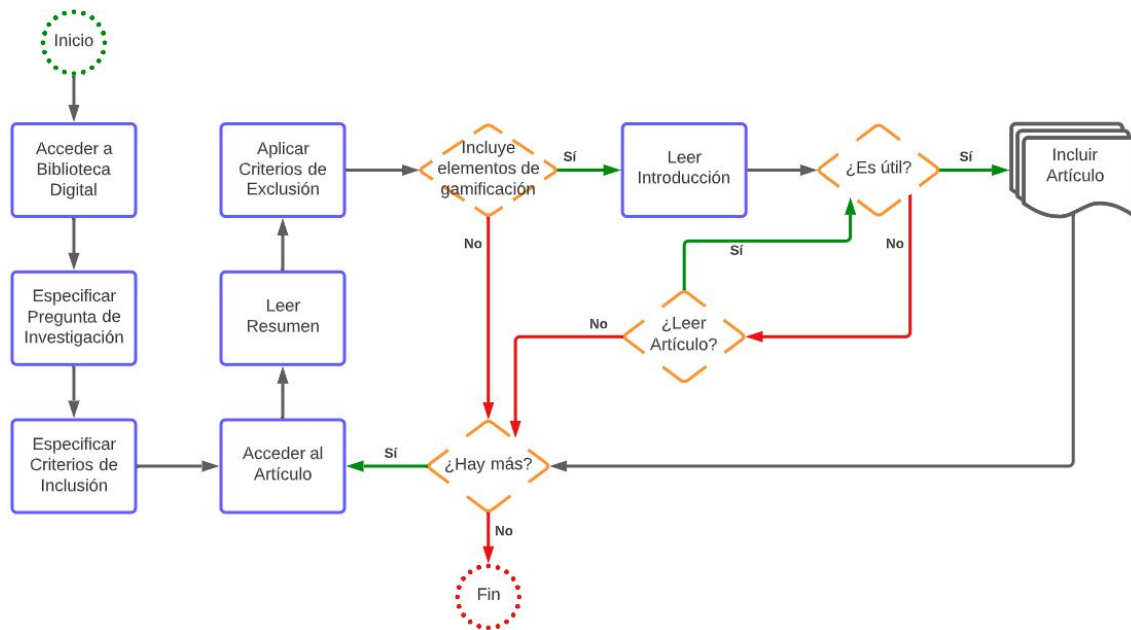


Figura 2.3: Proceso de selección de artículos

Con la información obtenida se procede a seleccionar la que se considera relevante para nuestro trabajo, utilizando para ello una tabla de evaluación donde se resume los artículos encontrados.

Se va a evaluar la calidad de cada artículo seleccionado para responder a la pregunta de investigación, por lo que se busca que los artículos contengan los siguientes conceptos: elementos de gamificación, técnicas de aprendizaje de la programación, juegos serios. Se comprueba si cada concepto aparece en los diferentes artículos, siendo el número de conceptos encontrados un indicador de la calidad del artículo en cuanto a si se adecua a lo que se busca y su utilidad para el trabajo. Con estos indicadores y tras la lectura del resumen se procede a calificar cada artículo, utilizando una puntuación entre 1 y 5, siendo 1 los artículos que menos conceptos contienen y cuyo resumen no hace hincapié en el uso de elementos gamificables, y 5 los que más.

En las imágenes 2.4, 2.5, 2.6 y 2.7 puede verse un resumen de las tablas creadas con información de los artículos revisados, guardando principalmente una referencia a la url donde aparece el artículo, la biblioteca que lo contiene y la puntuación otorgada tras la primera lectura.

Biblioteca	Nota [1-5]	Referencia	ACM	1	@inproceedings(10.1145/3285002.3285013,		
ACM	1	@article(10.1145/3394977,	ACM	1	@proceedings(10.1145/2897839,		
ACM	1	@inproceedings(10.1145/3364510.3366163,	ACM	1	@book(10.1145/2792790,		
ACM	1	@inproceedings(10.1145/2774225.2774834,	ACM	1	@proceedings(10.1145/2945078,		
ACM	1	@inproceedings(10.1145/2786805.2786806,	ACM	1	@reporte técnico(10.1145/3079760,		
ACM	1	@inproceedings(10.1145/3197768.3201561,	ACM	1	@inproceedings(10.1145/3344429.3372500,		
ACM	1	@inproceedings(10.1145/3462203.3475887,	ACM	1	@proceedings(10.1145/2995908,		
ACM	1	@inproceedings(10.1145/3059009.3059035,	ACM	1	@proceedings(10.1145/3359996,		
ACM	1	@inproceedings(10.1145/2617841.2617847,	ACM	1	@article(10.1145/3485128,	Springer	1 https://doi.org/10.1007/978-3-319-91152-6_26
ACM	1	@inproceedings(10.1145/2669711.2669907,	ACM	1	@proceedings(10.1145/3132787,	Springer	1 https://doi.org/10.1007/s10639-020-10296-1
ACM	1	@inproceedings(10.1145/3402942.3409617,	ACM	1	@proceedings(10.1145/3308560,	Springer	1 https://doi.org/10.1007/978-3-030-60128-7_28
ACM	1	@inproceedings(10.1145/3131085.3131115,	ACM	1	@book(10.1145/3015783,	Springer	1 https://doi.org/10.1007/978-3-030-21817-1_6
ACM	1	@inproceedings(10.1145/2818187.2818296,	ACM	1	@proceedings(10.1145/2993148,	Springer	1 https://doi.org/10.1007/978-3-642-39262-7_34
ACM	1	@inproceedings(10.5555/2667036.2667046,	IEEE Xplore	1	@INPROCEEDINGS(7942917,	Springer	1 https://doi.org/10.1007/978-3-319-94640-5_13
ACM	1	@inproceedings(10.1145/3485444.3487644,	Scopus	1	@article(SinAutor2020,	Springer	1 https://doi.org/10.1007/978-3-319-58515-4_15
ACM	1	@inproceedings(10.1145/3022636.3022648,	Scopus	1	@CONFERENCIA(SinAutor2018,	Springer	1 https://doi.org/10.1007/978-3-319-58509-3_36
ACM	1	@proceedings(10.1145/2729094,	Springer	1	https://doi.org/10.1007/978-3-030-60128-7_50	Springer	1 https://doi.org/10.1007/978-3-030-01842-9_1
ACM	1	@inproceedings(10.1145/3470482.3479615,	Springer	1	https://doi.org/10.1007/978-3-319-46152-6_4	Springer	1 https://doi.org/10.1007/978-3-030-77889-7_38
ACM	1	@article(10.1145/3411839,	Springer	1	https://doi.org/10.1007/978-3-030-11548-7_35	Springer	1 https://doi.org/10.1007/978-3-030-77857-6_28
ACM	1	@proceedings(10.1145/3134368,	Springer	1	https://doi.org/10.1007/978-3-319-70111-0_15	Springer	1 https://doi.org/10.1007/978-3-319-07222-2_26
ACM	1	@article(10.1145/2659796,	Springer	1	https://doi.org/10.1007/978-3-319-05972-3_9	Springer	1 https://doi.org/10.1007/978-3-030-77857-6_28
ACM	1	@article(10.1145/2888398,	Springer	1	https://doi.org/10.1007/978-3-319-51645-5_7	Springer	1 https://doi.org/10.1007/978-3-642-40477-1_38
ACM	1	@inproceedings(10.1145/3488162.3488163,	Springer	1	https://doi.org/10.1007/s10639-020-10213-4	Springer	1 https://doi.org/10.1007/978-3-030-72132-9_19
ACM	1	@article(10.1145/3015445.3015446,	Springer	1	https://doi.org/10.1007/978-3-030-77943-6_7	Springer	1 https://doi.org/10.1007/978-3-319-40216-1_35
ACM	1	@article(10.1145/2461256.2461275,	Springer	1	https://doi.org/10.1007/978-3-030-38778-5_17	Springer	1 https://doi.org/10.1007/978-3-030-77889-7_8
ACM	1	@article(10.1145/3460657,	Springer	1	https://doi.org/10.1007/978-3-030-15620-6_13	Springer	1 https://doi.org/10.1007/978-3-030-22602-2_21
ACM	1	@inbook(10.1145/2792790.2792815,	Springer	1	https://doi.org/10.1007/978-3-319-29060-7_1	Springer	1 https://doi.org/10.1007/978-3-319-67615-9_1
ACM	1	@proceedings(10.1145/2998181,	Springer	1	https://doi.org/10.1007/s10639-020-10396-w		
ACM	1	@inproceedings(10.1145/2441776.2441923,	Springer	1	https://doi.org/10.1007/978-3-030-34350-7_44		
ACM	1	@article(10.1145/2582193.2633443,	Springer	1	https://doi.org/10.1007/978-3-030-15620-6_4		
ACM	1	@proceedings(10.1145/2669062,	Springer	1	https://doi.org/10.1007/978-3-319-58625-0_2		
ACM	1	@proceedings(10.1145/3411763,	Springer	1	https://doi.org/10.1007/s40593-014-0016-x		
ACM	1	@inproceedings(10.1145/3285002.3285013,	Springer	1	https://doi.org/10.1007/978-1-4842-6287-0_4		
ACM	1	@inproceedings(10.1145/3285002.3285013,	Springer	1	https://doi.org/10.1007/978-3-319-20916-6_58		

Figura 2.4: Artículos con puntuación 1.

Biblioteca	Nota [1-5]	Referencia	Scopus	2	@CONFERENCIA(Buccharone2019753,		
ACM	2	@inproceedings(10.1145/3304221.3325593,	Springer	2	https://doi.org/10.1007/978-3-319-29585-5_24		
ACM	2	@inproceedings(10.1145/3362789.3362831,	Springer	2	https://doi.org/10.1007/978-3-030-15620-6_8	Springer	2 https://doi.org/10.1007/978-3-030-10576-1
ACM	2	@inproceedings(10.1145/3234152.3234161,	Springer	2	https://doi.org/10.1007/978-3-030-23712-7_8	Springer	2 https://doi.org/10.1007/978-3-319-40216-1_47
ACM	2	@inproceedings(10.1145/3364641.3364675,	Springer	2	https://doi.org/10.1007/978-3-319-51055-2_18	Springer	2 https://doi.org/10.1007/978-3-319-39513-5_41
ACM	2	@inproceedings(10.1109/AST.2017.20,	Springer	2	https://doi.org/10.1007/978-3-319-60013-0	Springer	2 https://doi.org/10.1007/978-3-030-01842-9_2
ACM	2	@inproceedings(10.1145/3468978.3468998,	Springer	2	https://doi.org/10.1007/s10639-020-10156-w	Springer	2 https://doi.org/10.1007/978-3-030-99188-3_24
ACM	2	@inproceedings(10.1145/2594776.2594778,	Springer	2	https://doi.org/10.1007/978-3-319-51645-5_29	Springer	2 https://doi.org/10.1007/s40593-017-0157-9
ACM	2	@inproceedings(10.1145/2828959.2828987,	Springer	2	https://doi.org/10.1007/978-3-319-46152-6_1	Springer	2 https://doi.org/10.1007/978-3-319-51645-5_4
ACM	2	@inproceedings(10.1145/2513002.2513024,	Springer	2	https://doi.org/10.1007/978-3-319-46152-6_10	Springer	2 https://doi.org/10.1007/s10639-021-10662-5
ACM	2	@inproceedings(10.1145/3472538.3472557,	Springer	2	https://doi.org/10.1007/978-3-030-11548-7_32	Springer	2 https://doi.org/10.1007/s10639-020-10276-2
ACM	2	@inproceedings(10.1145/3293881.3295779,	Springer	2	https://doi.org/10.1007/978-3-319-70111-0_28	Springer	2 https://doi.org/10.1007/s10639-021-10689-8
ACM	2	@inproceedings(10.1145/3350768.3352490,	Springer	2	https://doi.org/10.1007/978-3-319-39513-5_39	Springer	2 https://doi.org/10.1007/978-3-319-58509-3_34
ACM	2	@article(10.1145/3396249,	Springer	2	https://doi.org/10.1007/978-3-319-51645-5_14	Springer	2 https://doi.org/10.1007/978-3-030-15620-6_12
ACM	2	@inproceedings(10.1145/3099023.3099113,	Springer	2	https://doi.org/10.1007/978-3-319-45841-0_17	Springer	2 https://doi.org/10.1007/s40593-016-0126-8
ACM	2	@inbook(10.1145/3472410.3472430,	Springer	2	https://doi.org/10.1007/978-3-030-11548-7_21	Springer	2 https://doi.org/10.1007/978-3-319-52836-6_61
ACM	2	@inproceedings(10.1145/3364138.3364170,	Springer	2	https://doi.org/10.1007/978-3-030-15620-6_1	Springer	2 https://doi.org/10.1007/978-3-030-77943-6_5
ACM	2	@inproceedings(10.1145/3130859.3131302,	Springer	2	https://doi.org/10.1007/978-3-319-94640-5_15	Springer	2 https://doi.org/10.1007/978-3-030-22602-2_32
ACM	2	@inproceedings(10.1145/3338498.3358647,	Springer	2	https://doi.org/10.1007/978-3-319-40216-1_48	Springer	2 https://doi.org/10.1007/978-3-030-02762-9_35
ACM	2	@inproceedings(10.1145/3364138.3364161,	Springer	2	https://doi.org/10.1007/978-3-319-05972-3_16	Springer	2 https://doi.org/10.1007/978-3-030-50164-8_35
ACM	2	@inproceedings(10.1145/3230348.3230355,	Springer	2	https://doi.org/10.1007/978-3-030-34350-7_11	Springer	2 https://doi.org/10.1007/978-3-319-07440-5_60
ACM	2	@inproceedings(10.1145/3012430.3012650,	Springer	2	https://doi.org/10.1007/978-3-319-45841-0_2	Springer	2 https://doi.org/10.1007/978-3-319-78795-4_3
ACM	2	@inproceedings(10.1145/3502870.3506568,	Springer	2	https://doi.org/10.1007/978-3-030-22602-2_16	Springer	2 https://doi.org/10.1007/978-3-030-58796-3_12
ACM	2	@inproceedings(10.1145/3344429.3372507,	Springer	2	https://doi.org/10.1007/s10639-019-09968-2	Springer - Libro	2 https://doi.org/10.1007/978-3-030-10576-1
ACM	2	@inbook(10.1145/3399715.3399860,	Springer	2	https://doi.org/10.1007/978-3-030-11548-7_25	Springer - Libro	2 https://doi.org/10.1007/978-3-030-96060-5
ACM	2	@article(10.1007/s00779-020-01373-8,	Springer	2	https://doi.org/10.1007/s10639-020-10202-7	Springer - Libro	2 https://doi.org/10.1007/978-3-030-80504-3
ACM	2	@inproceedings(10.1145/2858796.2858801,	Springer	2	https://doi.org/10.1007/978-3-030-22602-2_25	Springer - Libro	2 https://doi.org/10.1007/978-3-030-49713-2
ACM	2	@article(10.1145/3234149,	Springer	2	https://doi.org/10.1007/978-3-030-50164-8_36		
ACM	2	@article(10.1145/3274444,	Springer	2	https://doi.org/10.1007/978-3-030-15620-6_7		
ACM	2	@inbook(10.1145/3015783.3015788,	Springer	2	https://doi.org/10.1007/978-3-658-07141-7_7		
IEEE Xplore	2	@INPROCEEDINGS(8820902,	Springer	2	https://doi.org/10.1007/978-3-319-60013-0		

Figura 2.5: Artículos con puntuación 2.

Biblioteca	Nota [1-5]	Referencia
ACM	3	@inproceedings{10.1145/3341215.3356335,
ACM	3	@inproceedings{10.1145/3286606.3286851,
ACM	3	@inproceedings{10.1145/2645791.2645813,
ACM	3	@inproceedings{10.1145/3134368.3139220,
ACM	3	@inproceedings{10.1145/3012430.3012661,
ACM	3	@article{10.1145/3412843,
ACM	3	@inproceedings{10.1145/3408877.3432544,
ACM	3	@inproceedings{10.1145/3322645.3322695,
ACM	3	@inproceedings{10.1145/3422392.3422497,
ACM	3	@inproceedings{10.1145/2702123.2702552,
ACM	3	@inproceedings{10.1145/3175326.3175576,
ACM	3	@inproceedings{10.1145/2677758.2677760,
ACM	3	@inbook{10.1145/3383668.3419888,
ACM	3	@article{10.1145/3365000,
ACM	3	@inproceedings{10.1145/3059009.3059052,
ACM	3	@inproceedings{10.1145/3436756.3437024,
ACM	3	@inproceedings{10.1145/2669711.2669895,
ACM	3	@inproceedings{10.1145/2491411.2494589,
ACM	3	@inproceedings{10.1145/33197091.33197123,
ACM	3	@inproceedings{10.1145/3337722.3337743,
ACM	3	@inproceedings{10.1145/3090354.3090451,
ACM	3	@inproceedings{10.1145/2658537.2658683,
ACM	3	@inproceedings{10.1145/3368089.3417932,
ACM	3	@inbook{10.1145/3410404.3414253,
ACM	3	@inproceedings{10.1145/2503859.2503881,
ACM	3	@inbook{10.1109/MODELOS-C.2019.00117,
ACM	3	@book{10.1145/2625848,
ACM	3	@inproceedings{10.1145/2559206.2567814,
ACM	3	@article{10.1145/3474656,
ACM	3	@inproceedings{10.1145/3483899.3483910,
ACM	3	@inproceedings{10.1145/2984393.2984394,
ACM	3	@inproceedings{10.1145/3402942.3403016,
ACM	3	@inproceedings{10.1145/3039677.3039681,
ACM	3	@inproceedings{10.1145/2808580.2808650,
ACM	3	@inproceedings{10.1145/3284179.3284295,
ACM	3	@inproceedings{10.1145/3123818.3123858,
ACM	3	@inproceedings{10.1145/3411170.3411251,
ACM	3	@article{10.5555/3381569.3381581,
ACM	3	@inproceedings{10.1145/3297662.3265802,
ACM	3	@inproceedings{10.1145/3422392.3422490,
ACM	3	@inproceedings{10.1145/2858796.2858797,
IEEE Xplore	3	@INPROCEEDINGS{8899715,
IEEE Xplore	3	@INPROCEEDINGS{9245291,
IEEE Xplore	3	@INPROCEEDINGS{8618195,
IEEE Xplore	3	@INPROCEEDINGS{8658937,
IEEE Xplore	3	@INPROCEEDINGS{8564214,
Scopus	3	@CONFERENCIA{Zhaoc2021,
Scopus	3	@article{Li2021216,
Springer	3	https://doi.org/10.1007/s10639-017-9622-1
Springer	3	https://doi.org/10.1007/978-3-030-58459-7_29
Springer	3	https://doi.org/10.1007/978-3-319-71940-5_14
Springer	3	https://doi.org/10.1007/s10639-020-10228-x
Springer	3	https://doi.org/10.1007/978-3-642-33687-4_4
Springer	3	https://doi.org/10.1007/s10639-020-10295-8
Springer	3	https://doi.org/10.1007/s10639-021-10459-6
Springer	3	https://doi.org/10.1007/978-3-319-29585-5_16
Springer	3	https://doi.org/10.1007/978-3-319-60013-0_36
Springer	3	https://doi.org/10.1007/978-3-319-46152-6_8
Springer	3	https://doi.org/10.1007/978-3-030-61814-8_11
Springer	3	https://doi.org/10.1007/978-3-030-58459-7_30
Springer	3	https://doi.org/10.1007/978-3-319-39483-1_29
Springer	3	https://doi.org/10.1007/978-3-030-23513-0_14
Springer	3	https://doi.org/10.1007/978-3-319-46152-6_5
Springer	3	https://doi.org/10.1007/978-3-030-34350-7_36
Springer	3	https://doi.org/10.1007/s10639-021-10502-6
Springer	3	https://doi.org/10.1007/s10639-021-10668-z
Springer	3	https://doi.org/10.1007/978-3-030-23513-0_15
Springer	3	https://doi.org/10.1007/978-3-030-72132-9_6
Springer	3	https://doi.org/10.1007/s10639-021-10717-7
Springer	3	https://doi.org/10.1007/978-3-319-40216-1_9
Springer	3	https://doi.org/10.1007/978-3-030-58459-7_21
Springer	3	https://doi.org/10.1007/978-3-319-39483-1_42
Springer	3	https://doi.org/10.1007/978-3-319-40216-1_49
Springer	3	https://doi.org/10.1007/978-3-030-42156-4_1
Springer	3	https://doi.org/10.1007/s10639-021-10488-1
Springer - Libro	3	https://doi.org/10.1007/978-3-030-77889-7
Springer - Libro	3	https://doi.org/10.1007/978-3-030-88272-3
Springer - Libro	3	https://doi.org/10.1007/978-3-030-86439-2
Springer - Libro	3	https://doi.org/10.1007/978-3-030-78095-1
Springer - Libro	3	https://doi.org/10.1007/978-3-030-98404-5

Figura 2.6: Artículos con puntuación 3.

Como puede verse en la imagen 2.7, la tabla utilizada también contiene información como el DOI de los artículos puntuados con 4 y 5, y aunque no se vea en la imagen también se ha utilizado para incluir algunas palabras clave que hagan recordar a los artículos, o un pequeño resumen.

Biblioteca	Nota [1-5]	Referencia
ACM	5	@inproceedings{10.1145/3287324.3287480,
ACM	5	@inproceedings{10.1145/3287324.3287411,
ACM	5	@inproceedings{10.1145/2583008.2583015,
ACM	5	@inproceedings{10.1145/2669711.2669904,
ACM	5	@inproceedings{10.1145/3364641.3364661,
ACM	5	@inproceedings{10.1145/3019612.3019752,
IEEE Xplore	5	@INPROCEEDINGS{8312377,
IEEE Xplore	5	@INPROCEEDINGS{9125283,
Springer	5	https://doi.org/10.1007/978-3-319-51645-5_21
Springer	5	https://doi.org/10.1007/978-3-319-71940-5_11
Springer	5	https://doi.org/10.1007/978-3-030-77857-6
ACM	4	@inproceedings{10.1145/3025453.3025711,
ACM	4	@article{10.1145/3231709,
ACM	4	@inproceedings{10.1145/3266003.3266007,
ACM	4	@inproceedings{10.1145/2669711.2669902,
ACM	4	@article{10.1145/3427597,
ACM	4	@inproceedings{10.1109/ICSE-SEET.2017.17,
ACM	4	@inproceedings{10.1145/3284179.3284290,
ACM	4	@inproceedings{10.1145/3001773.3001832,
ACM	4	@inproceedings{10.1145/2538862.2538921,
ACM	4	@inproceedings{10.1145/2538862.2538887,
IEEE Xplore	4	@INPROCEEDINGS{9354285,
IEEE Xplore	4	@INPROCEEDINGS{9274194,
IEEE Xplore	4	@INPROCEEDINGS{9354271,
IEEE Xplore	4	@INPROCEEDINGS{8850951,
IEEE Xplore	4	@INPROCEEDINGS{8756848,
IEEE Xplore	4	@INPROCEEDINGS{8988818,
IEEE Xplore	4	@INPROCEEDINGS{8956286,
Scopus	4	@CONFERENCIA{Figueiredo2020997,
Scopus	4	@article{Vera201855,
Scopus	4	@CONFERENCIA{Barik2016134,
Springer	4	https://doi.org/10.1007/978-3-030-21151-6_17
Springer	4	https://doi.org/10.1007/978-3-319-93843-1_1
Springer	4	https://doi.org/10.1007/978-3-319-51645-5_22
Springer	4	https://doi.org/10.1007/978-3-319-20609-7_56
Springer	4	https://doi.org/10.1007/978-3-319-91152-6_13
Springer	4	https://doi.org/10.1007/978-3-319-19126-3_5
Springer	4	https://doi.org/10.1007/s10639-018-9730-6
Springer	4	https://doi.org/10.1007/s10639-013-9291-7
Springer - Libro	4	https://doi.org/10.1007/978-3-030-99188-3
Springer - Libro	4	https://doi.org/10.1007/978-3-030-77943-6
Springer - Libro	4	https://doi.org/10.1007/978-3-030-86436-1
Springer - Libro	4	https://doi.org/10.1007/978-3-030-78270-2

Figura 2.7: Artículos con puntuación 4 y 5.

Una vez obtenidas las puntuaciones se observa que aparecen 11 artículos con calificación 5, y 32 artículos con calificación 4, decidiendo descartar el resto de puntuaciones. Tras leer todos los artículos se decide descartar los que no incluían información relevante

sobre los elementos de gamificación. En la tabla 2.6 puede verse el número de trabajos encontrados y los que finalmente han sido seleccionados por cada biblioteca digital.

Bibliotecas	Encontrados	Útiles
ACM Digital Library	133	16
IEEE Xplore	16	9
Springer Link	149	15
Scopus Preview	9	3

Tabla 2.6: Resultados de las búsquedas

2.2 Ejecución y análisis de resultados

La ejecución de la revisión se realizó entre febrero y mayo de 2022. Se comenzó filtrando los artículos encontrados según el contenido del título, el resumen, y en algunos casos la introducción. Seguidamente se realizó una revisión algo más detallada de los artículos obtenidos, analizando sus contenidos para conseguir así obtener los artículos definitivos a utilizar en el mapeo sistemático. A continuación se procedió a descartar artículos que no aportaban información relevante, realizando una revisión completa de los artículos resultantes como útiles para el estudio, recalcando que la adecuación de los artículos al trabajo de investigación no tienen relación con la calidad de los artículos, sino con su utilidad y nivel de detalle a la pregunta de investigación.

Durante la revisión se analizan los conceptos presentes en los artículos relacionados con los elementos de gamificación, viendo que no siempre se profundiza en ellos. Por ejemplo, hay trabajos que simplemente mencionan los elementos utilizados, sin detallar los motivos por los que se usan.

Una vez realizada la revisión, se procede a analizar la información obtenida sobre los elementos de gamificación. En la tabla 2.7 se identifican los elementos presentes en los artículos revisados, teniendo en cuenta que algunos nombran los elementos, otros los identifican y otros los incluyen en sus propuestas. También hemos encontrado trabajos que se centran en algunos elementos en concreto.

Artículo	Niveles	Barra Progreso	Sanciones	Límite Tiempo	Clasificación	Insignias	Objetivos	Trofeo-Recompensa	Puntuación	Bien Virtual	Comentarios	Asistencia	Libertad Fallar	Avatares	Multijugador	Misión-Desafío
[34]		x			x	x			x	x	x		x	x		
[35]	x				x			x							x	
[6]	x		x			x	x	x			x	x		x	x	
[9]																
[36]	x	x			x	x		x	x	x	x			x	x	
[23]	x	x			x	x		x	x	x				x	x	x
[20]	x	x			x	x		x	x	x				x	x	x
[42]	x				x	x			x		x					x
[21]	x	x			x	x		x	x							
[18]	x	x			x	x		x	x		x	x		x	x	
[46]		x			x	x		x	x		x	x		x		x
[19]	x						x									x
[32]		x			x	x		x	x		x				x	x
[30]	x				x		x		x	x				x		x
[8]	x	x			x	x					x			x	x	
[48]	x	x		x	x				x						x	x
[26]					x	x										
[29]	x	x			x	x		x	x		x				x	
[17]					x	x			x		x			x	x	
[10]	x			x		x			x							
[40]	x	x			x			x							x	x
[43]	x	x		x	x				x			x				
[44]	x	x			x	x			x	x				x		x
[33]	x				x	x	x		x							
[39]					x	x		x	x		x					
[37]	x	x		x	x			x	x					x		
[15]	x				x	x		x	x							
[38]	x	x			x	x	x		x	x				x		
[27]		x			x			x	x				x			
[24]	x						x				x					
[13]	x	x			x	x			x	x	x	x		x		x
[31]	x				x	x	x		x						x	
[12]					x		x		x			x		x	x	
[28]	x	x	x			x	x	x			x	x				x
[14]	x	x			x	x			x						x	
[49]	x	x			x	x	x		x							
[50]	x	x		x			x						x	x	x	
[22]	x	x			x				x					x		

Tabla 2.7: Elementos presentes en los artículos

El elemento que incluye un límite de tiempo puede tener un efecto negativo, ya que puede provocar que el alumno se centre más en finalizar el juego que en adquirir conocimiento.

2.3 Respuestas a la pregunta de investigación

Una vez finalizada la revisión de los artículos, se realiza un análisis de los elementos encontrados en los 42 artículos, mostrando en la imagen 2.8 una gráfica con el número de apariciones de los elementos comunes en los trabajos del estudio, observando que son varios los elementos que superan las 21 apariciones, que corresponde a la mitad de los artículos revisados. Se tendrá en cuenta que en la gráfica se muestran los elementos que más veces aparecen, ya que se nombran más elementos pero en contadas ocasiones.

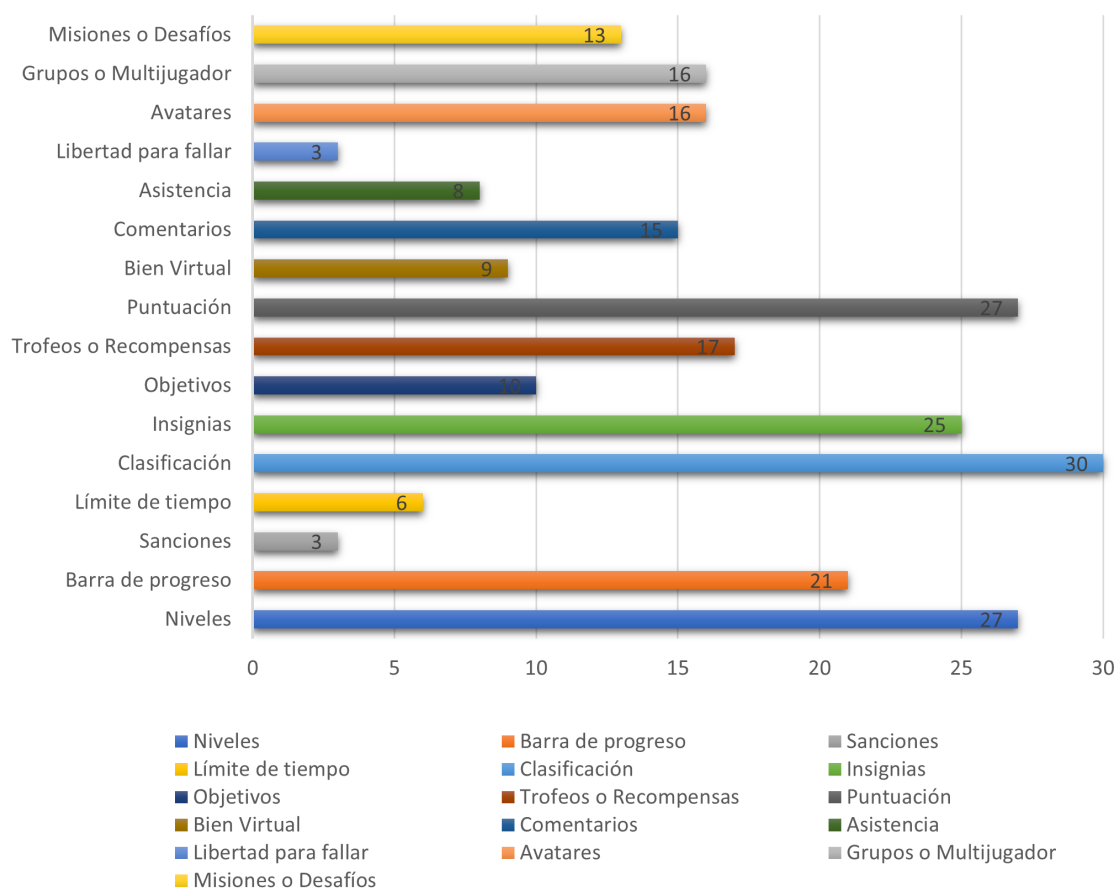


Figura 2.8: Número de apariciones de los elementos en los artículos

Seguidamente de los 16 elementos de gamificación estudiados se extrae el top-10, resumido en el gráfico 2.9. A continuación se listan los elementos por el número de veces que aparecen en los artículos, siendo el primero el que más veces se nombre, y el último el que menos:

1. Clasificación
2. Niveles
3. Puntuación
4. Insignias

5. Barra de progreso
6. Trofeos o recompensas
7. Avatares
8. Grupos o multijugador
9. Comentarios
10. Misiones o desafíos

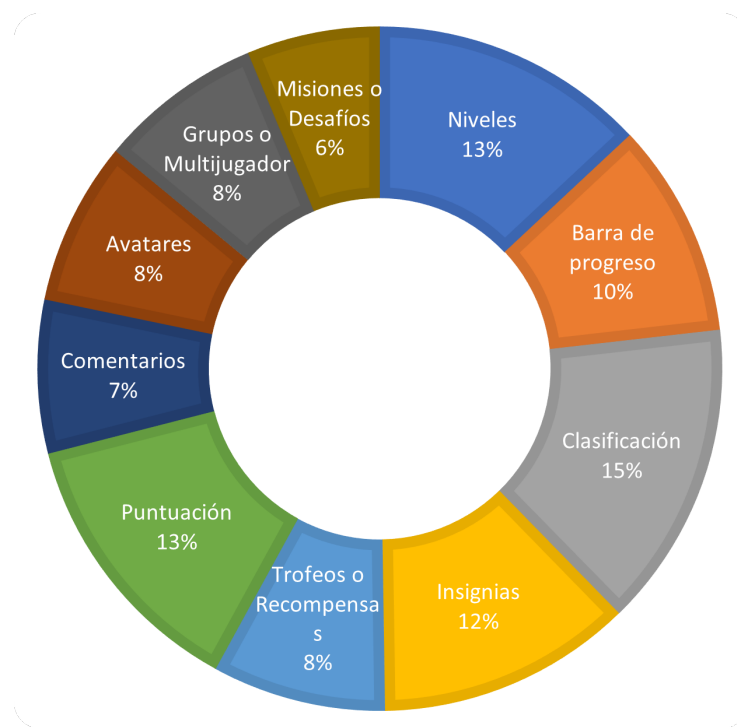


Figura 2.9: Top-10 de elementos presentes en los artículos

Todos estos elementos de gamificación serían candidatos a ser incluidos en el juego, ya que puede considerarse que influirán positivamente en el uso del juego. Algunos de ellos ya aparecen en el juego, como son los niveles y los comentarios, por lo que se ha incluido un nuevo nivel con su correspondiente narrativa, utilizando además diferentes idiomas para que pueda entenderse mejor según la lengua materna del jugador.

Otros elementos han sido descartados al estar algo alejados del estilo actual del juego, y no tener una puntuación alta, como son los avatares, los grupos-multijugador, y las misiones-desafíos. Estos podrían considerarse para un trabajo futuro.

El resto de elementos han sido considerados como adecuados para tratar de mejorar el juego, y han tratado de ser incluidos de diferentes maneras. Por ejemplo para las recompensas se pueden conseguir diferentes objetos al ir superando niveles, los cuales pueden ser utilizados para personalizar el personaje principal del juego. Otro elemento incluido sería la barra de progreso, donde de un vistazo el jugador puede ver todos los niveles y los que ha superado. El método de puntuación utilizado sería el número de niveles superados, el cual puede verse al ir remarcando en la barra de progreso cada nivel en el que el jugador consiga llevar al robot a la meta. También se podrán conseguir insignias al superar un número de niveles.

Han quedado fuera de las mejoras elementos como los objetivos, los bienes virtuales, la asistencia, el límite e tiempo, la libertad para fallar y las sanciones. En los artículos revisados también aparecen otros elementos de gamificación, sin llegar a repetirse en varias ocasiones ninguno de ellos.

CAPÍTULO 3

Diseño del juego

En esta sección se revisan los diferentes niveles del juego, enumerando los elementos que incluye y los objetivos de aprendizaje que pretenden conseguir. El orden de los niveles ha sido fijado según el orden de contenidos del curso de programación de la Universidad Politécnica de Valencia, ya que tienen muchos años de experiencia en la enseñanza de la programación.

En el juego se han incluido nuevos elementos de gamificación, los cuales son una barra de navegación, premios al superar niveles que permiten la personalización del robot, una puntuación por niveles superados, nuevas acciones para detener el movimiento del robot, una tabla de clasificación donde pueden verse los nombres de los alumnos que más niveles han superado, ayuda a la narración incluyendo más idiomas, insignias por superar niveles, y un premio final.

Se tendrá en cuenta que el objetivo del juego es llevar al robot desde el punto inicial del tablero hasta la casilla final, utilizando para ello las instrucciones y mecanismos que van aprendiendo en los diferentes niveles. En el tablero aparecen obstáculos que deben evitar para conseguir que el robot avance.

3.1 Elementos comunes en todos los niveles

En la zona superior se permite seleccionar el idioma en el que se mostrará el juego y las instrucciones, como puede verse en 3.1. A los que ya disponía se le han incluido el valenciano, portugués, alemán y japonés. El poder utilizar el idioma habitual del alumno ayuda a que asimile mejor los conocimientos.



Figura 3.1: Barra de idiomas

En la imagen 3.2 puede verse cómo aparecería el escenario del juego al seleccionar el idioma valenciano, modificando los títulos, los textos de los botones y la explicación del nivel, teniendo en cuenta que en caso de aparecer algún aviso informativo también aparece traducido.

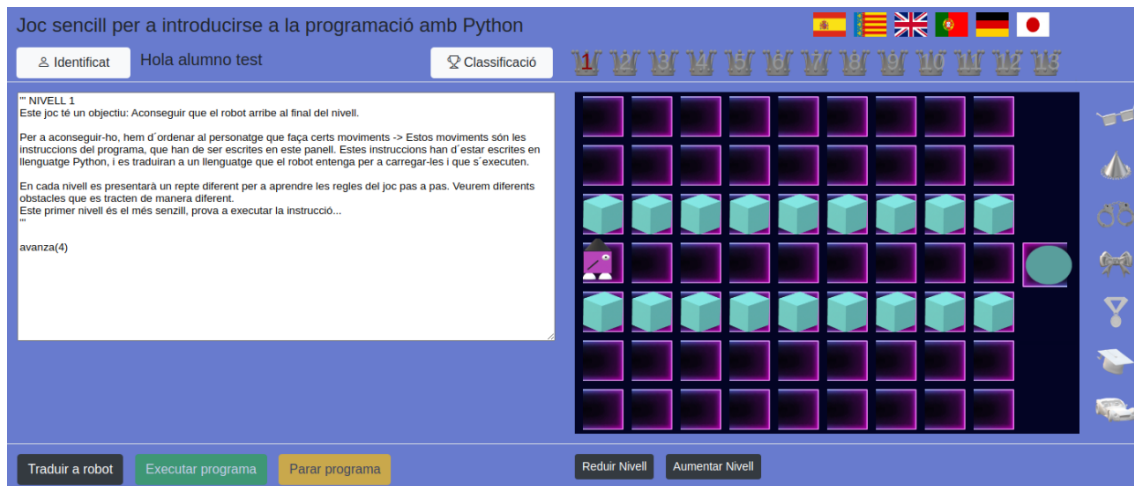


Figura 3.2: Juego en valenciano

Para incluir nuevos idiomas de una forma más cómoda se ha rediseñado el código del juego para utilizar un mapeo con los diferentes idiomas, diferenciando cada expresión con un identificador. En la imagen 3.3 puede verse un ejemplo de definición de idiomas.

```
map_esp.set('avanza', "avanza");
map_esp.set('gira', "gira");
map_esp.set('titulo', "Juego sencillo para introducir a la programación en Python");
map_esp.set('btnTraducir', "Traducir a robot");
map_esp.set('btnEjecutar', "Ejecutar programa");
map_esp.set('btnParar', "Parar programa");

map_eng.set('avanza', "forward");
map_eng.set('gira', "turn");
map_eng.set('titulo', "Easy game to introduce programming in Python");
map_eng.set('btnTraducir', "Traslate to robot");
map_eng.set('btnEjecutar', "Execute program");
map_eng.set('btnParar', "Stop program");

map_vlc.set('avanza', "avança");
map_vlc.set('gira', "gira");
map_vlc.set('titulo', "Joc senzill per a introduir-se a la programació amb Python");
map_vlc.set('btnTraducir', "Traduir a robot");
map_vlc.set('btnEjecutar', "Executar programa");
map_vlc.set('btnParar', "Parar programa");
```

Figura 3.3: Ejemplo de mapeo de idiomas

En caso de incluir un nuevo idioma, debe definirse una variable en el archivo de mapeos de idiomas, incluyendo todas las expresiones con su correspondiente traducción. Seguidamente, en el código debe especificarse el nuevo idioma, asignando el nombre del mapeo y las expresiones regulares que va a utilizar.

Otra zona común es la barra de progreso y navegación, la cual aparece encima del mapa del robot, como puede verse en la imagen 3.4. Permite conocer el nivel actual mostrarse el número del nivel resaltado en color rojo, marcando con color verde el número y coloreando la corona del nivel que ya ha sido superado. También permite navegar a los diferentes niveles al hacer clic en el nivel deseado.



Figura 3.4: Barra de progreso y navegación

En la zona superior se irán mostrando las insignias que va consiguiendo el alumno según el número de niveles que supera 3.5. Al superar 4 elementos se le otorga la primera insignia, cuando llega a 8 la segunda, y al llegar a los 12 la tercera.



Figura 3.5: Insignias que puede conseguir el alumno

En el lateral derecho aparecen los premios que puede ir consiguiendo el alumno al superar los diferentes niveles 3.6. Inicialmente aparecerán coloreados en gris, y al ir consiguiendo los premios pasarán a tener color, permitiendo ponérselos al robot. Los premios se agrupan según el lugar del cuerpo del robot donde se coloca, pudiendo ser la cabeza, el cuerpo o los pies. El robot únicamente puede tener puesto un objeto de cada parte del cuerpo. Se tendrá en cuenta que al hacer clic sobre un premio ganado, si este lo lleva puesto el robot se le quitará, y si vuelve a hacer clic se lo volverá a poner.



Figura 3.6: Barra de premios

A través del botón **Identifícate** el alumno puede indicar su nombre o alias, mostrando un formulario con el campo nombre, como puede verse en 3.7. Al pulsar el botón **OK** el nombre indicado será mostrado junto a un saludo.

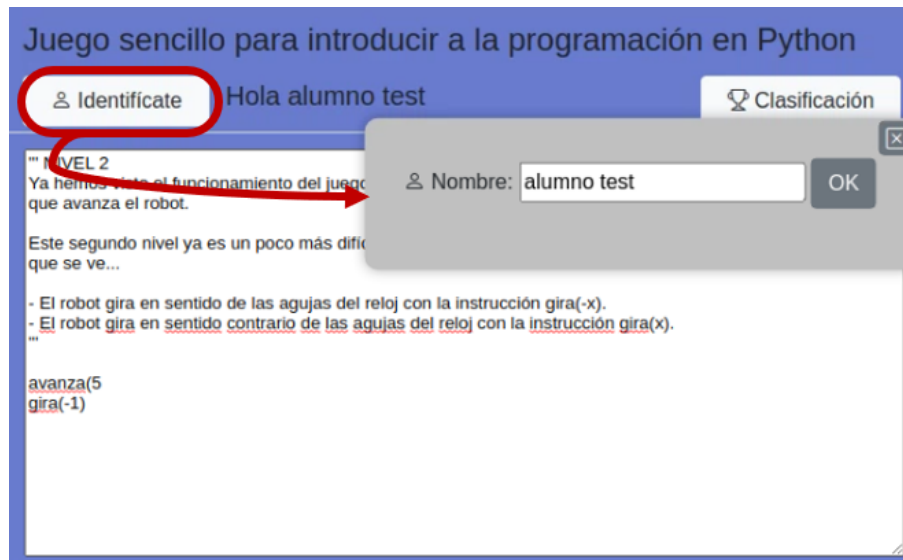


Figura 3.7: Formulario de identificación del alumno

Otro elemento común es el botón **Clasificación**. Este botón muestra los 10 primeros clasificados según el número de niveles superados y la fecha en la que se superen, siendo mejor clasificado el que la fecha sea más reciente. En 3.8 puede verse el panel que se muestra tras pulsar el botón, donde aparece una tabla con la información de la clasificación.

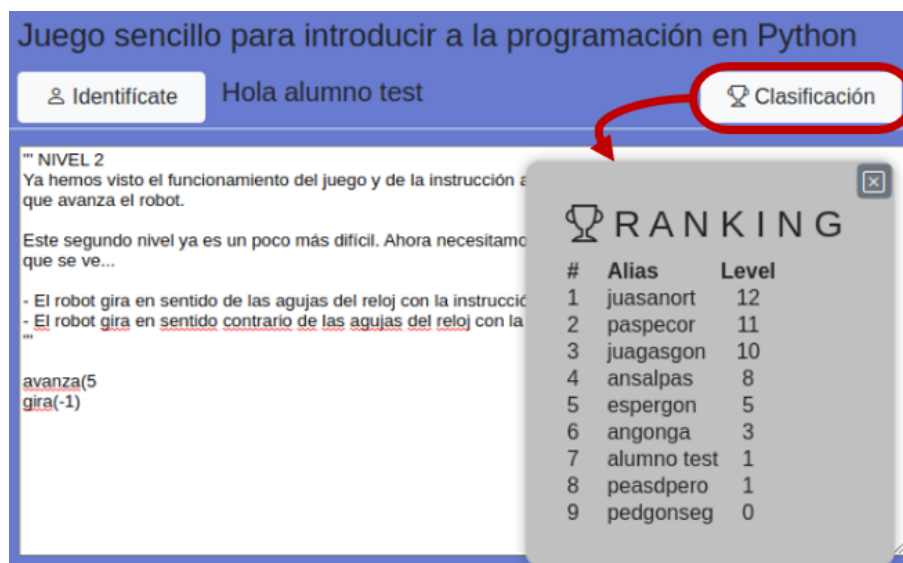


Figura 3.8: Formulario con la clasificación

3.2 Nivel 1

En el primer nivel se introduce el juego. Aparece el tablero con el robot y los obstáculos que marcan por donde no puede pasar. Se muestra un panel con una explicación del propósito a aprender en este nivel, junto a un ejemplo de instrucción, como puede verse en 3.9.



Figura 3.9: Nivel 1

Aquí pueden verse algunos de los nuevos elementos de gamificación incluidos, como la barra de progreso y navegación, los diferentes idiomas disponibles, y los premios que permiten personalizar al robot. Estos elementos son comunes en los diferentes niveles.

El alumno debe ser capaz de modificar la instrucción **avanza**, especificando entre los paréntesis el número de casillas por las que el robot va a moverse para llegar al final. Seguidamente debe pulsar el botón **Traducir a robot**, donde en caso de haber escrito una instrucción que no sea válida se le avisará a través del panel inferior. Si la instrucción es válida se activará el botón **Ejecutar programa**, el cual debe pulsar para que el robot inicie los movimientos que se han definido en las instrucciones. Como puede observarse los obstáculos mostrados marcan claramente el camino a seguir, haciendo que el alumno se centre en analizar la instrucción y en cómo conseguir que se traduzca y ejecute a través de las acciones permitidas en el juego.

Al superar este primer nivel el alumno consigue un premio, en este caso son unas gafas, mostrando un aviso informativo. Se permite utilizar los premios conseguidos a través del panel lateral, donde al realizar clic sobre un premio el robot se pondrá el objeto pulsado, quitándose en caso de tenerlo puesto previamente. También destacar que la corona del primer nivel pasa de ser gris a tener color dorado, y que el número del nivel ahora es verde, indicando así que lo ha superado. En 3.10 se destacan estos elementos de gamificación incorporados al juego.

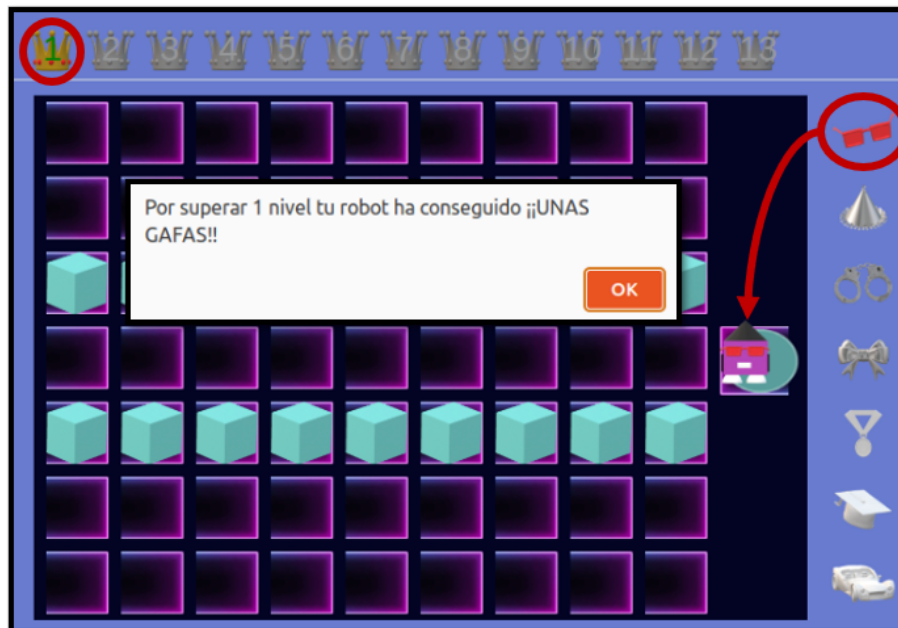


Figura 3.10: Premio de unas gafas al superar 1 nivel

3.3 Nivel 2

Si el alumno está en el nivel 1 puede pasar al nivel 2 a través del botón *Aumentar nivel*, mientras que si está en cualquier otro nivel puede acceder seleccionando la corona con el número 2. Las coronas muestran de una manera sencilla al alumno el nivel en el que está actualmente, ya que aparece el número en color rojo. También permite ver el número de niveles que han ido superando al aparecer la corona de color dorado y el número en color verde, actuando como una barra de progreso que además permite saltar a los diferentes niveles, como puede verse en la imagen 3.11, donde se destacan la corona conseguida del nivel 1, y la corona 2 del nivel actual.



Figura 3.11: Nivel 2

En este nivel se introduce una nueva instrucción, lo cual produce que aumente la dificultad al buscar la solución. También aparecen menos obstáculos, haciendo que la solución sea más abierta, y al aparecer un obstáculo en la parte central obliga a utilizar la instrucción **gira** introducida en este nivel, la cual el alumno debe entender para conseguir que el robot se oriente en la dirección deseada y así poder avanzar hacia el destino final. En los comentarios del panel lateral se explica la nueva instrucción, mostrando un ejemplo de uso, el cual puede traducir y ejecutar para ver el comportamiento del robot. En este nivel se hace ver al alumno que puede utilizar varias instrucciones consecutivas, las cuales se ejecutan de forma secuencial, y que el parámetro que se indica en cada instrucción hace que el robot se comporte de diferente manera.

3.4 Nivel 3

Este nivel incluye un nuevo obstáculo con pinchos, el cual no permite continuar en caso de chocar con él. En caso de superar este nivel se consigue un nuevo premio, un gorro de fiesta, como puede verse en la imagen 3.12. Al tener varios premios se pueden combinar para que el robot pueda utilizarlos, al estilo de Mr. Potato.



Figura 3.12: Premio de un gorro de fiesta al superar 3 niveles

En la imagen 3.12 también puede observarse que las coronas coloreadas ya son 3, lo cual indica de una forma visual y sencilla el avance que el jugador está realizando.

Este nivel introduce el obstáculo con pinchos, el cual provoca que el robot pare la ejecución en caso de chocar con él, haciendo ver al alumno que al programar pueden darse situaciones más restrictivas que otras, ya que no se continuará con la secuencia de instrucciones.

3.5 Niveles 4 y 5

Estos niveles introducen un nuevo concepto, se trata de mostrar obstáculos en diferentes posiciones del mapa, donde en caso de volver a cargar el nivel 4 o el 5 se mostrarán de forma aleatoria en una posición o en otra algunos de los obstáculos, como puede verse en la imagen 3.13.

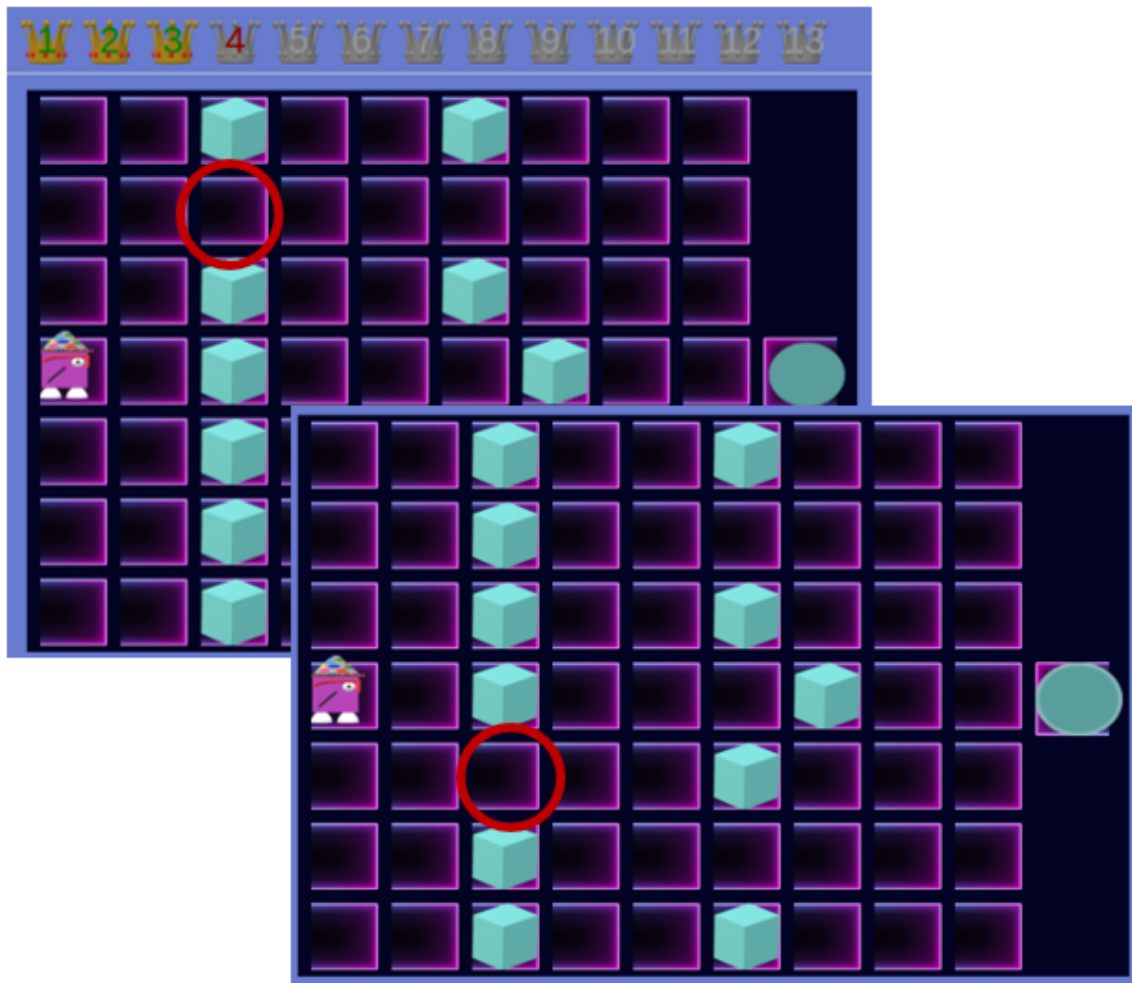


Figura 3.13: Nivel 4 con obstáculo aleatorio

A través de estos 2 niveles se pretende que el alumno practique las instrucciones, permitiendo que busque diferentes soluciones según las posiciones de los obstáculos cargadas al mostrar cada uno de estos niveles, buscando que asimile los conocimientos.

También se tendrá en cuenta que al superar 4 niveles se consigue la primera de las tres insignias, la cual aparecerá en la zona superior del juego, como puede verse en la imagen 3.14.

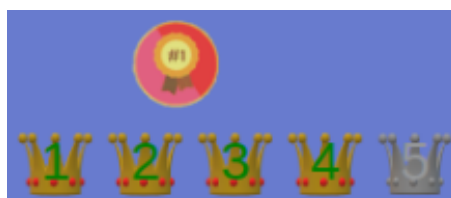


Figura 3.14: Insignia conseguida al superar 4 niveles

3.6 Nivel 6

Este nivel no incluye nuevos conceptos, sino que plantea un mapa diferente, donde únicamente puede llegarse a la meta siguiendo el camino superior o inferior, ya que el central obligaría a ir hacia atrás. Al superar el nivel, si el alumno ha conseguido que sea

el sexto, se consigue como premio unas esposas, las cuales ahora dejan de estar en gris para aparecer en rojo, como puede verse en 3.15, donde también se ve como se colorean los 6 niveles conseguidos.



Figura 3.15: Nivel 6, consiguiendo como premio unas esposas

Llegados a este nivel parece un buen momento para que el alumno test revise las posiciones en la tabla de clasificación, donde como se muestra en la imagen 3.16 está en la 5ª posición con 6 niveles superados.

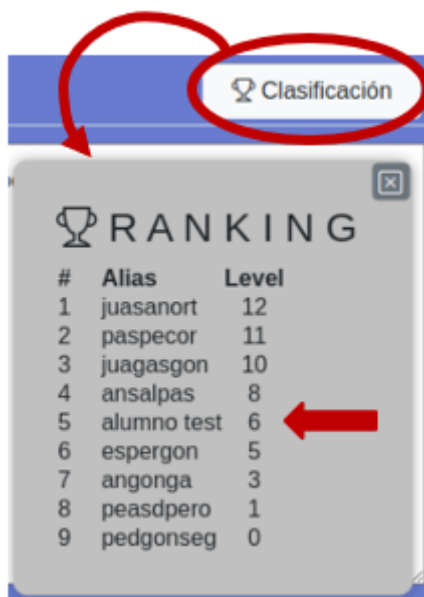


Figura 3.16: Clasificación tras superar 6 niveles

3.7 Nivel 7

Este nivel hasta ahora aparecía en la posición 9, y se ha modificado por el 7 para seguir el temario impartido en asignaturas de programación de la Universitat Politècnica de València, ya que introduce la instrucción condicional, la cual se ejecuta sólo si se cumple la condición. Se trata de un nivel donde se hace uso también de las posiciones aleatorias de los obstáculos, dificultando un poco más el encontrar la solución.

El alumno debe comprender lo que se explica en los comentarios del panel izquierdo, ya que es importante seguir la nomenclatura de la instrucción `si` que incluye el carácter `:` y la indentación, características de la programación en Python.

Aparece por primera vez una imagen con una brújula, como puede verse en 3.17, la cual se mostrará en los niveles posteriores, siendo útil para que el alumno se ubique al utilizar los parámetros `robot_sur`, `robot_este`, `robot_norte` y `robot_oeste` de los condicionales.



Figura 3.17: Nivel 7, incluyendo la brújula

En caso de superar este nivel se acumularían 7, obteniendo como premio un lazo, como puede verse en la imagen 3.18. El lazo, al ubicarse en la cabeza obliga a que no pueda utilizarse el gorro de fiesta, el cual ahora el alumno puede intercambiar por el lazo haciendo clic en su imagen.



Figura 3.18: Lazo como premio al superar 7 niveles

3.8 Nivel 8

El nivel 8 antes se correspondía con el 10. Este nivel no aporta nada nuevo, pero es uno de los niveles más difícil de superar al ser las ubicaciones de los obstáculos con pinchos completamente aleatorias, excepto la celda que da paso a la meta, ya que en ese caso sería imposible de superar el nivel, como se muestra en 3.19.



Figura 3.19: Nivel 8

Al superar este nivel se consigue llegar a 8 superados, obteniendo una segunda insignia, como puede verse en 3.20.



Figura 3.20: Segunda insignia conseguida al superar 8 niveles

3.9 Nivel 9

Introduce el concepto de bucle repetitivo con una condición, el cual ejecuta las instrucciones que contiene mientras la condición se cumpla. Este nivel antes aparecía en la posición 7, siendo modificado para adaptarse al currículum actual de asignaturas de programación de la UPV. En 3.21 puede verse la distribución de los obstáculos, donde en caso de ejecutar el código propuesto como ejemplo el alumno observará que el robot gira alrededor del primer obstáculo de la derecha.



Figura 3.21: Nivel 9

Se tendrá en cuenta que a partir de este nivel es útil el botón **Parar programa**, el cual se encarga de detener la ejecución tras finalizar la instrucción que actualmente está aplicando el robot, no siguiendo con la siguiente de las instrucciones de la secuencia

programada. Su utilidad radica en la necesidad de parar la ejecución en caso de repetir el bucle en un gran número de ocasiones, viendo que no se consigue avanzar en la solución, ya que en ese caso el robot repetiría los movimientos hasta que se cumpliera la condición.

Al superar el nivel, si se acumulan 9 niveles superados se consigue un premio, siendo en este caso una medalla, como se observa en la imagen 3.22. Al disponer de varios objetos el alumno puede jugar a ver que ta le sientan los premios conseguidos al robot.



Figura 3.22: Medalla como premio al superar 9 niveles

3.10 Nivel 10

Este nivel es completamente nuevo, e introduce la instrucción **mientras** y la condición **no_choca**. En caso de probar a traducir y ejecutar las instrucciones propuestas el alumno debe darse cuenta que el robot continuará la ejecución infinitamente, ya que gira a la izquierda y avanza 2 posiciones mientras no encuentre oposición. Ante esta situación es cuando puede hacer uso del botón **Parar programa**, aunque como medida preventiva también se avisa que existe un número máximo de repeticiones configurado a 50.



Figura 3.23: Nivel 10

Al estar en una fase del juego donde se han ido superando muchos niveles, si se consigue superar el décimo ya se premia con un birrete, como puede verse en la imagen 3.24. El alumno está progresando en la práctica y esto se hace notar en los premios y coronas activados, dando colorido a la interfaz gráfica.



Figura 3.24: Medalla como premio al superar 10 niveles

3.11 Nivel 11

El nivel 11 antes se correspondía con el 8. Se trata de un nivel completamente aleatorio, como puede verse en 3.25, donde los huecos de las columnas de obstáculos varían cada vez que se recarga el nivel. Aquí el alumno puede practicar haciendo uso de todas las instrucciones que se han ido adquiriendo durante el avance en el juego.



Figura 3.25: Nivel 11

3.12 Nivel 12

Este nivel introduce el concepto de testeo de código, como puede verse en 3.26. Antes aparecía como el nivel 11, siendo la última de las instrucciones introducidas en el juego.



Figura 3.26: Nivel 12

Al conseguir 12 niveles se premia con un coche y la tercera y última insignia, como puede verse en la imagen 3.27. Con este premio se consigue completar todos los objetos que pueden ser utilizados para personalizar el robot.



Figura 3.27: Coche al superar 12 niveles, y la tercera insignia

3.13 Nivel 13

Último nivel, es el de mayor dificultad al tener obstáculos que obligan a que la solución utilice muchas instrucciones. Todas las columnas de obstáculos son aleatorias, como puede verse en 3.28, buscando que el alumno pueda practicar a programar ante diferentes situaciones del mapa.



Figura 3.28: Nivel 13

En caso de superar este último nivel, si ya había superado todos los anteriores, se consigue un premio final, que consiste en una animación de confeti, tal y como muestra la imagen 3.29.

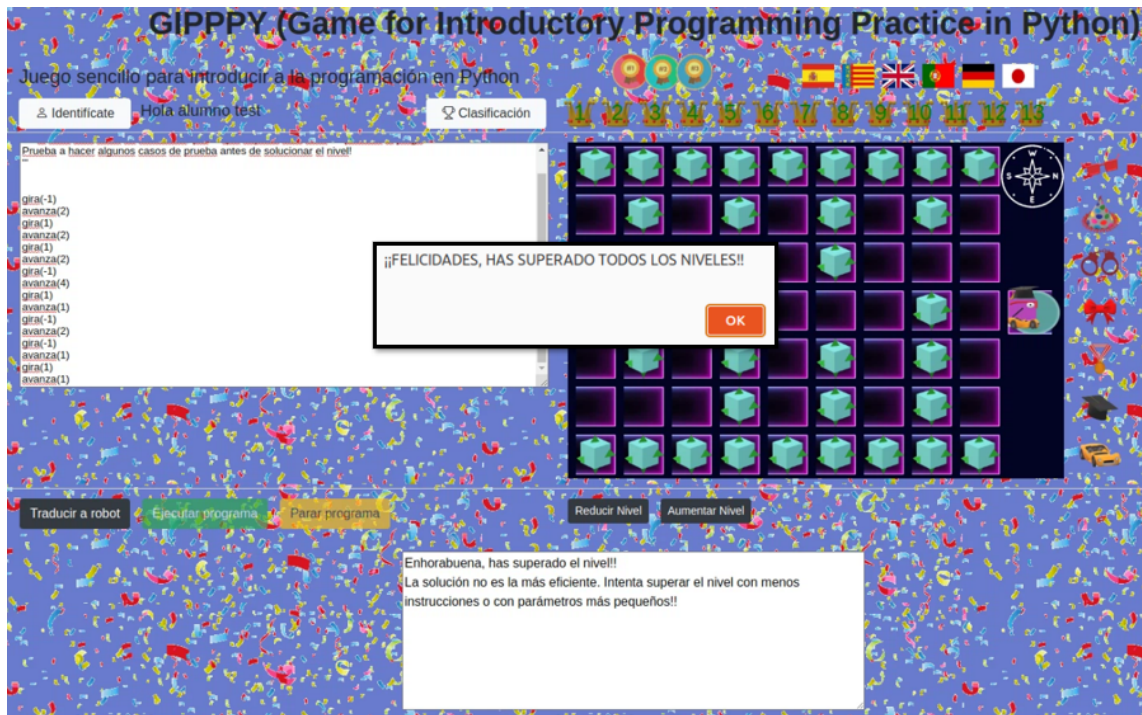


Figura 3.29: Confeti final al superar todos los niveles

Llegado a este punto es un buen momento para ver la clasificación en la que ha quedado el alumno, como puede verse en 3.30

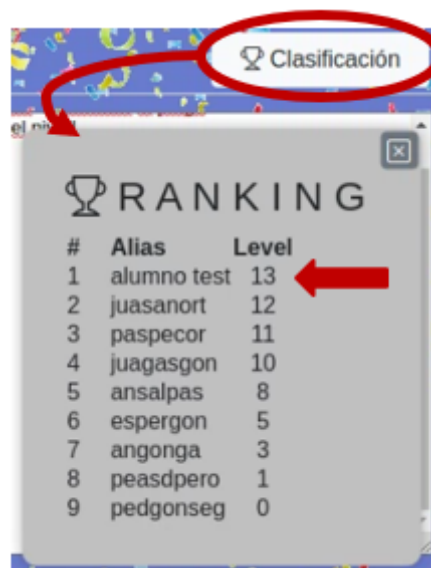


Figura 3.30: Clasificación tras superar 13 niveles

CAPÍTULO 4

Diseño del software

Este capítulo detalla el diseño del software entendido como el proceso de planificación y creación de la estructura, comportamiento y apariencia visual del juego. Es un paso importante en el desarrollo de cualquier aplicación, ya que un diseño bien planteado ayuda a que la aplicación sea funcional, fácil de usar y de mantener.

Al hablar del diseño de una aplicación software suelen nombrarse conceptos clave como la arquitectura, la interfaz de usuario y la funcionalidad.

4.1 Requisitos

La aplicación tiene que cumplir una serie de requisitos software, los cuales se detallan a continuación:

- Contar con un juego online para practicar Python.
- Tener niveles que permitan practicar las instrucciones, condiciones IF, ciclos (repite y mientras) y diseño de casos de prueba.
- Tener niveles aleatorios que permitan añadir dinamismo al juego si se quiere jugar más de una vez.
- Incluir conceptos generales de programación traducción/compilación, la ejecución/parada del programa.
- Añadir los elementos de gamificación más utilizados o recomendados en la literatura.
- Entregar feedback al alumno cada vez que ejecuta un programa.

Mantener la calidad es otro de los requisitos a cumplir en cualquier aplicación web para que se ofrezca una buena experiencia de usuario, y que sea fácil de mantener. Para conseguirlo se ha revisado el código para que al incluir nuevas funcionalidades se mantenga la estructura definida, refactorizando algunas partes del código para que sea más legible y mantenible, y realizado pruebas unitarias tras incluir cada funcionalidad y pruebas de integración al finalizar el desarrollo para que la experiencia al jugar siga siendo la adecuada.

4.2 Arquitectura

La arquitectura de una aplicación se refiere a su estructura general, incluyendo cómo se organizan y se comunican los diferentes módulos de la aplicación. Gippy inicialmente se implementó en Javascript combinado con HTML y CSS, y ha pasado a utilizar Django con Python y una base de datos SQLite. La imagen 4.1 resume su arquitectura actual.

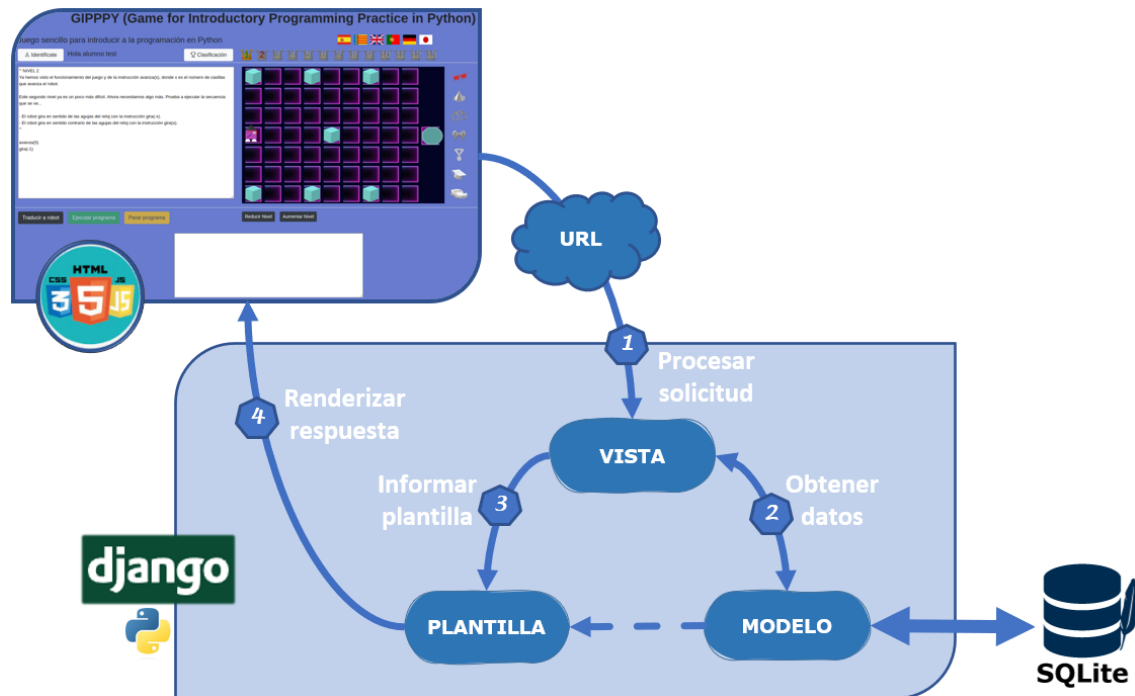


Figura 4.1: Arquitectura general de la aplicación

A continuación se detallan cada una de las partes que conforman la arquitectura:

- Interfaz web:** Para desarrollar el lado cliente se ha hecho uso de HTML, CSS y JavaScript. Estos tres elementos se encargan de dar forma a la parte frontal de GIPPPY. HTML se utiliza para crear la estructura básica de la página web, incluyendo los botones, mapa de juego, imágenes, formularios, cuadros de texto y etiquetas. CSS se utiliza para dar estilo y diseño, controlando la apariencia de los elementos HTML, como el color, la fuente, el tamaño y la ubicación, teniendo en cuenta que algunos de estos elementos hacen uso de clases de Bootstrap. JavaScript se utiliza para definir la funcionalidad y la comunicación con el servidor.
- Django¹:** Se trata de un marco de desarrollo web de código abierto para Python ampliamente utilizado para el desarrollo de aplicaciones web. Es muy versátil y permite crear aplicaciones web complejas con una gran variedad de funcionalidades, incluyendo el almacenamiento de información en una base de datos. Su arquitectura se basa en el patrón Modelo-Vista-Controlador, utilizando una combinación de modelos, vistas, plantillas y enrutamiento para crear aplicaciones web seguras y escalables de manera rápida y eficiente. Django incluye soporte para la integración con bases de datos relacionales, utilizando un mapeo Objeto-Relacional (ORM) que permite trabajar con bases de datos utilizando objetos Python en lugar de escribir consultas SQL directamente.

¹<https://www.djangoproject.com/>

- **SQLite²**: Es la base de datos utilizada. Se trata de un sistema de gestión de bases de datos relacionales de código abierto, ligero y autónomo. Su integración con Django es sencilla, ya que inicialmente debe definirse la conexión a la base de datos, y para trabajar se pueden crear modelos que representan las tablas, utilizando consultas ORM para interactuar con los datos.

Para el desarrollo del servicio web con Django se ha utilizado el IDE para Python *PyCharm*³, el cual dispone de una versión gratuita "PyCharm Community Edition". Esta versión es totalmente funcional y ofrece la mayoría de las características que un desarrollador necesita para escribir, depurar y ejecutar código Python 4.2.

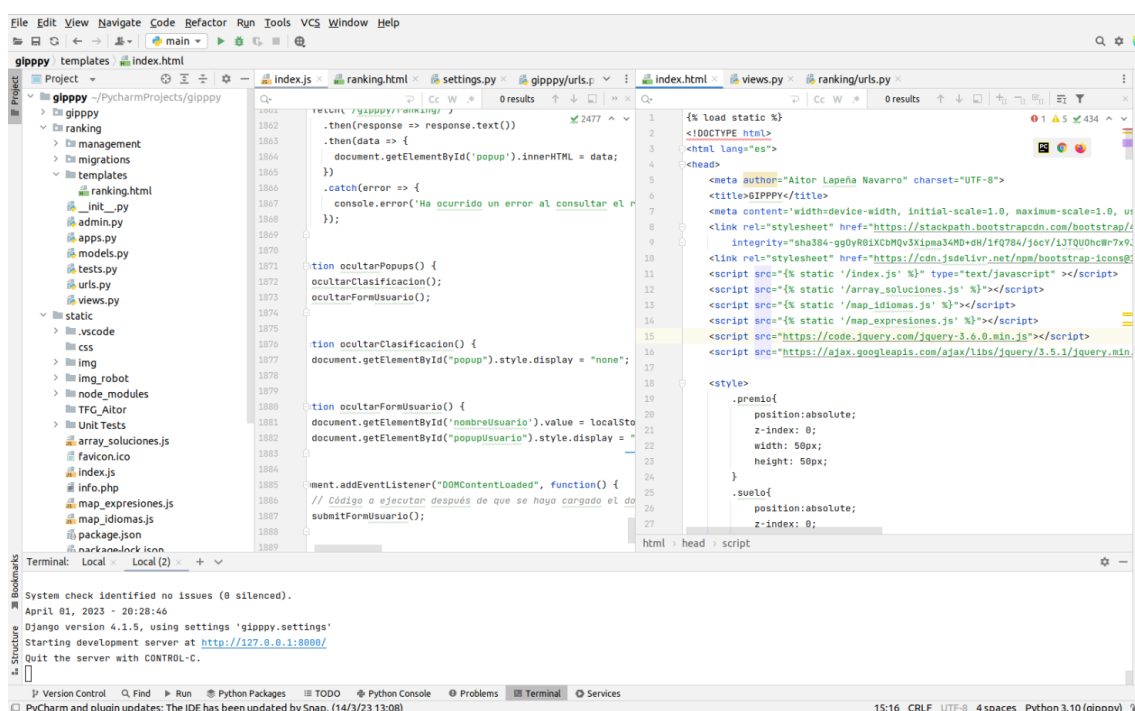


Figura 4.2: IDE para Python Pycharm

Para gestionar la base de datos se ha utilizado la herramienta de código abierto *DB Browser for SQLite (DB4S)*⁴, la cual permite trabajar de forma visual con bases de datos SQLite 4.3. Permite realizar las gestiones habituales en bases de datos SQLite, como crear y eliminar tablas, ejecutar consultas SQL, editar, añadir y eliminar registros, importar y exportar datos, entre otras. Además ofrece una interfaz gráfica fácil de usar que permite visualizar los datos en forma de tablas y gráficos, lo que facilita la comprensión y el análisis de la información almacenada en la base de datos, haciendo que durante las fases de desarrollo y pruebas sea puedan gestionar los datos de forma sencilla y rápida.

²<https://www.sqlite.org/>

³<https://www.jetbrains.com/es-es/pycharm-edu/>

⁴<https://sqlitebrowser.org/>

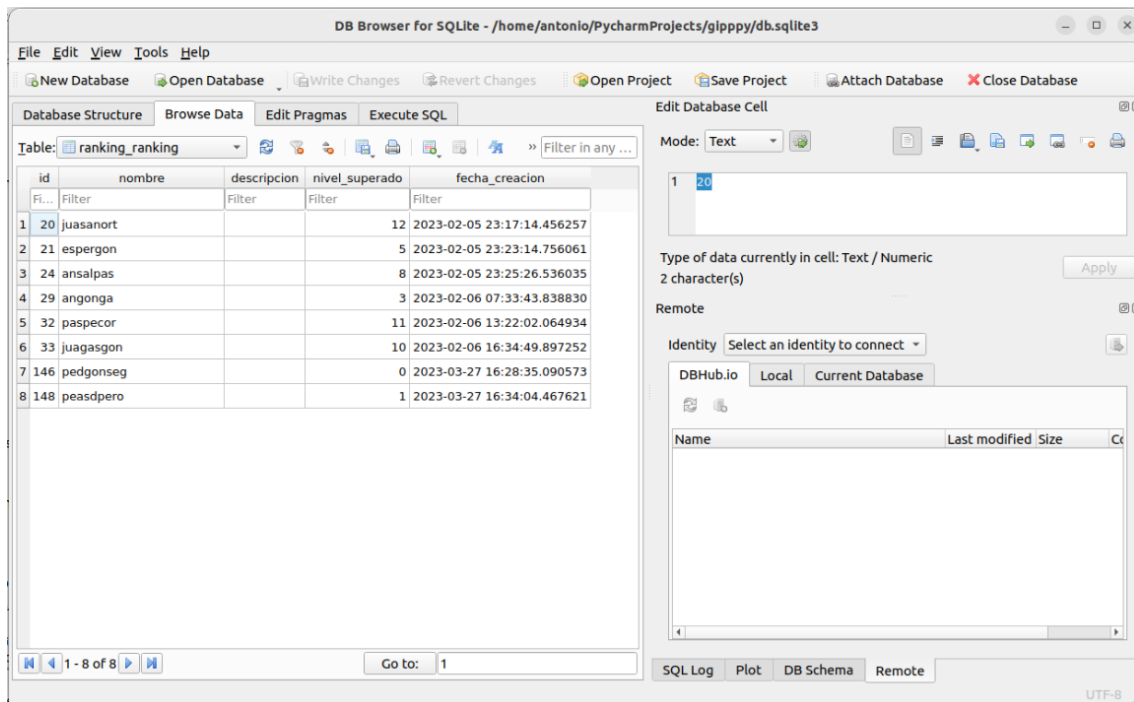


Figura 4.3: Herramienta utilizada para administrar bases de datos SQLite

En cuanto a los issues encontrados, uno de ellos era la seguridad al comunicar entre el cliente y el servidor, ya que es un aspecto crítico en las aplicaciones web. Para garantizar la seguridad en la comunicación con el servidor Django hace uso del middleware *SecurityMiddleware*, incluyendo encabezados de seguridad a las respuestas HTTP enviadas por la aplicación, ayudando a protegerse contra ataques comunes.

Además, para protegerla de ataques malintencionados CSRF (Cross-Site Request Forgery) donde se envían peticiones no autorizadas, se hace uso de la protección CSRF de Django, añadiendo un token al HTML, el cual es comparado con el almacenado en el servidor, rechazando la petición en caso de no coincidir.

Otra posible acción a realizar es definir una cuota máxima para el fichero que contiene la base de datos SQLite, consiguiendo así limitar el tamaño para que en caso de un ataque malintencionado no se llene el disco.

También se ha preparado en Django el comando personalizado incluido en el fichero *limpiar_rankings_antiguos.py* 4.1, el cual permite borrar registros con fecha de creación anterior a 3 meses, por lo que podría prepararse un cron que lo ejecute.

```

1 from django.core.management.base import BaseCommand
2 from ranking.models import Ranking
3 from datetime import date, datetime
4
5
6 class Command(BaseCommand):
7     help = 'Mantener unicamente los rankings de los ultimos 3 meses de la base
8         de datos'
9
10    def handle(self, *args, **options):
11        fecha_borrar = date.today() - datetime.timedelta(days=90)
12        ranking_antiguos = Ranking.objects.filter(fecha_creacion__lt=
13            fecha_borrar)
14        ranking_antiguos.delete()
15        self.stdout.write(self.style.SUCCESS('Se eliminaron los rankings
16            anteriores a 3 meses'))

```

Listing 4.1: Comando personalizado en Django

Otra posibilidad sería que el comando permita parámetros para especificar que se quiere borrar, como por ejemplo definir el número de días a mantener, o que borre los que no estén entre los X primeros clasificados, y que se lanzara diariamente.

4.3 Interfaz de usuario

La interfaz de usuario es el medio a través del cual los usuarios interactúan con la aplicación. Se ha tratado de mantener su diseño al incluir los elementos de gamificación que afectan a la interfaz, como son las imágenes que permiten personalizar el robot a través de los premios por superar niveles, los idiomas disponibles a través de banderas, la barra de navegación, las insignias, los botones para la tabla de clasificación, la identificación del usuario y la posibilidad de parar la ejecución del programa, tal y como como puede verse en la imagen 4.4.

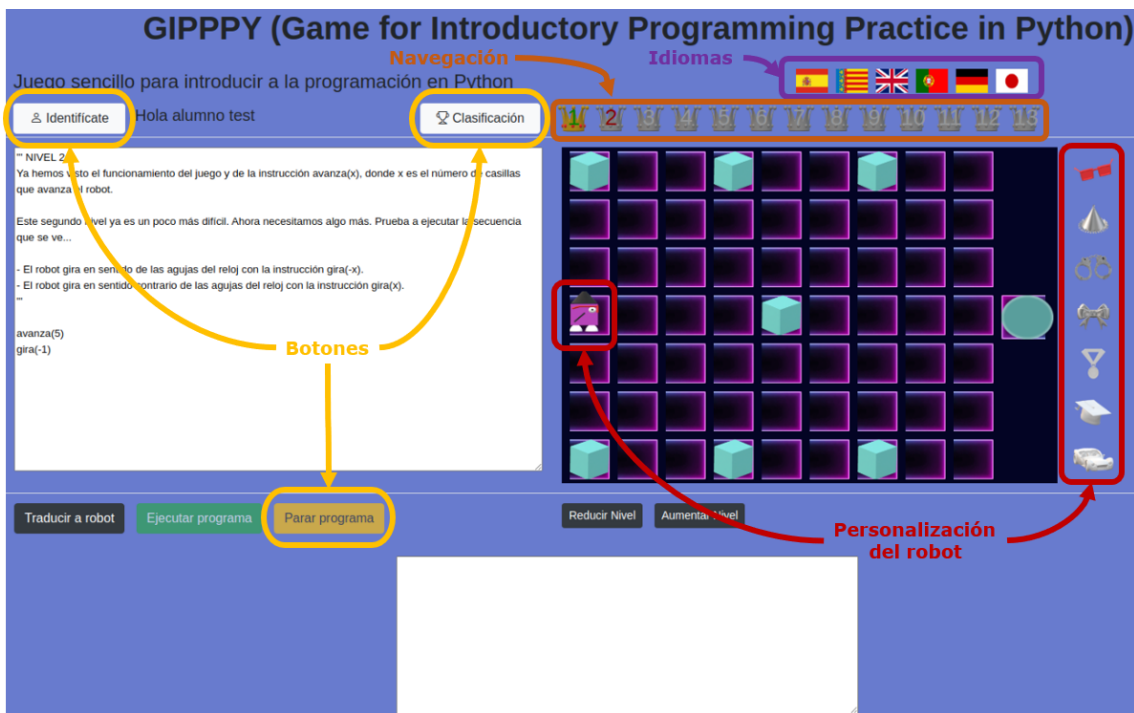


Figura 4.4: Interfaz gráfica

CAPÍTULO 5

Implementación

En este capítulo se explican las ampliaciones incluidas en la funcionalidad del juego, las modificaciones realizadas para incluir nuevos elementos de gamificación y la refactorización del código aplicada.

Inicialmente al desarrollar GIPPY se ha hecho uso de Visual Studio Code¹, un editor de código fuente muy popular desarrollado por Microsoft que está disponible de forma gratuita para Windows, macOS y Linux. Es un software de código abierto y se utiliza principalmente para programar aplicaciones y sitios web, debido a que cuenta con una gran cantidad de extensiones y complementos que permiten incluir nuevas funcionalidades al editor, como su integración con Git, además de ser compatible con una amplia gama de lenguajes de programación, entre los cuales están JavaScript, HTML y CSS. Otra característica importante es su capacidad para depurar código, la cual ha sido muy útil en la fase de conocimiento del código del juego para trazar los pasos que sigue, y en la corrección de errores durante el desarrollo. En la imagen 5.1 puede verse la apariencia de este IDE.

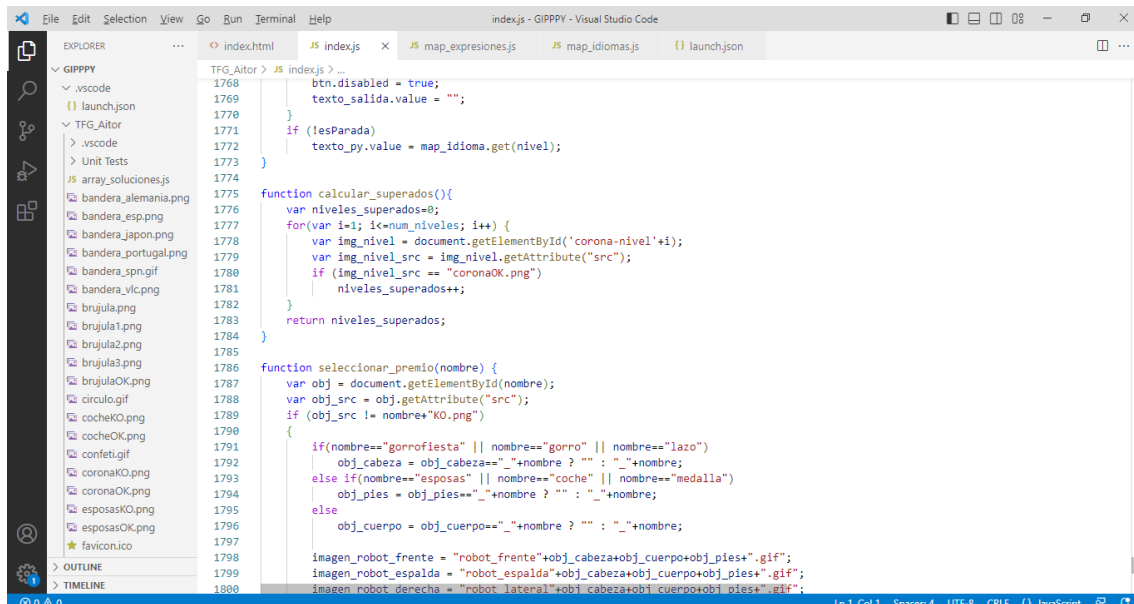


Figura 5.1: IDE Visual Studio Code

También se ha hecho uso de las DevTools para Chrome, como puede verse en la imagen 5.2. Se trata de una serie de herramientas para desarrolladores integradas en el ex-

¹<https://code.visualstudio.com/>

plorador web google Chrome, a través de las cuales se permite depurar y optimizar el código fuente de aplicaciones web. Ha sido de utilidad para revisar el código HTML al integrar la parte gráfica de los elementos de gamificación incluidos en GIPPPY, ayudando además a depurar el código JavaScript en las funcionalidades desarrolladas.

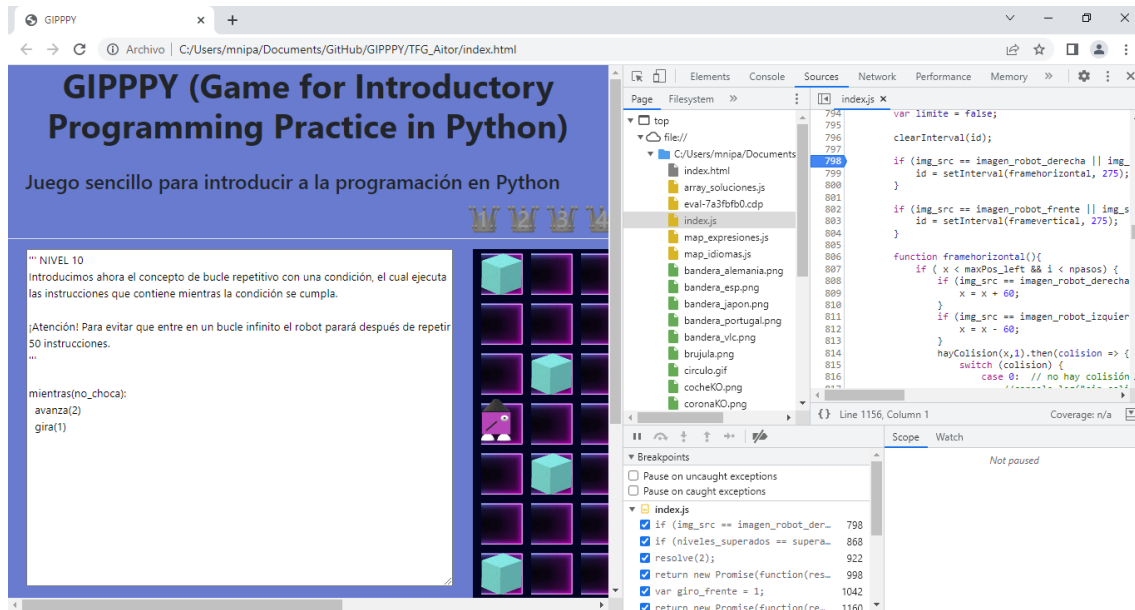


Figura 5.2: Herramientas para desarrolladores del Chrome

En la fase desarrollo en la que se incluyeron los premios que permiten personalizar el personaje principal fue necesario el uso de una herramienta para la edición de imágenes, optando por utilizar principalmente el editor de gráficos Microsoft Paint 3D, el cual está incluido a partir del sistema operativo Windows 10. En la imagen 5.3 pueden verse los diferentes premios, los cuales se han ido combinando con las imágenes del robot para conseguir que aparezcan en todas las perspectivas desde las que puede verse.

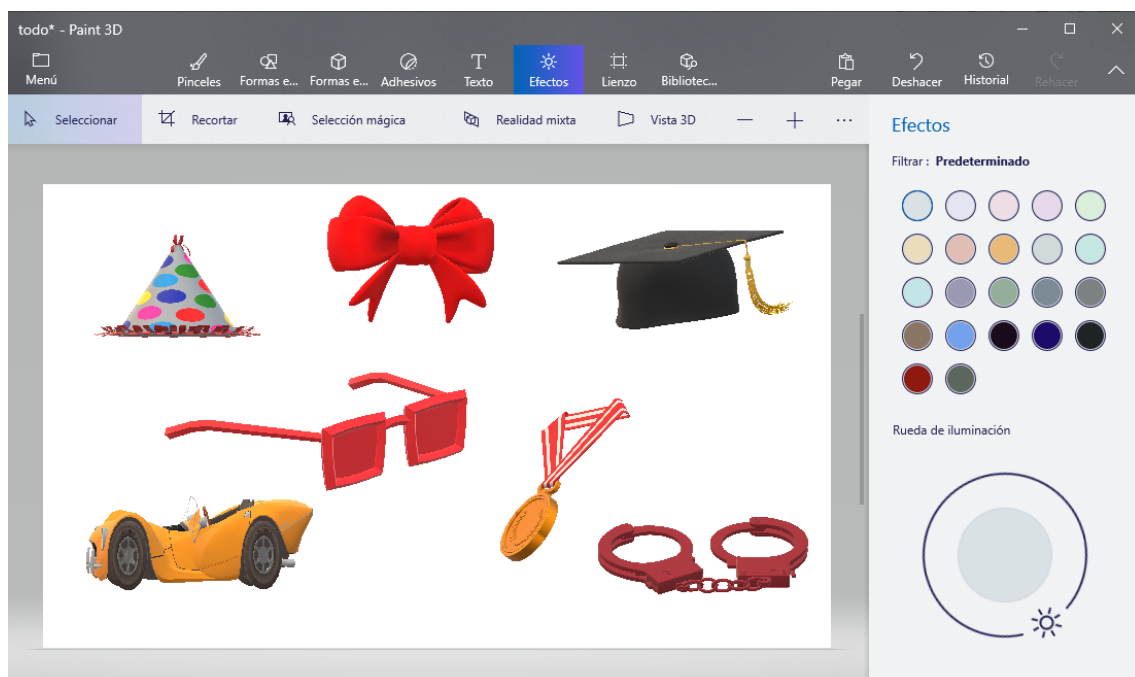


Figura 5.3: Editor de gráficos Microsoft Paint 3D

En la fase final del desarrollo del presente trabajo es cuando se pasó a utilizar Py-Charm debido a las facilidades que otorga para trabajar con el framework Django. Este IDE ha sido presentado anteriormente en 4.2, pudiendo verse un ejemplo de su interfaz en la imagen 4.2.

Respecto al repositorio donde almacenar el código fuente, se ha mantenido el utilizado en anterior trabajo en GitHub², utilizando GitHub Desktop para gestionarlo. Para mantener las primeras modificaciones del código se decide utilizar otra rama, cuyo nombre es *alara*, incluyendo comentarios explicando el contenido de los cambios realizados en cada uno de los commits. Para las modificaciones que incluyen el uso del framework Django, al ser un cambio significativo en cuanto a estructura del código, se utiliza la rama *django*, la cual incluye todos los cambios realizados en el proyecto, como puede verse en la imagen 5.4.

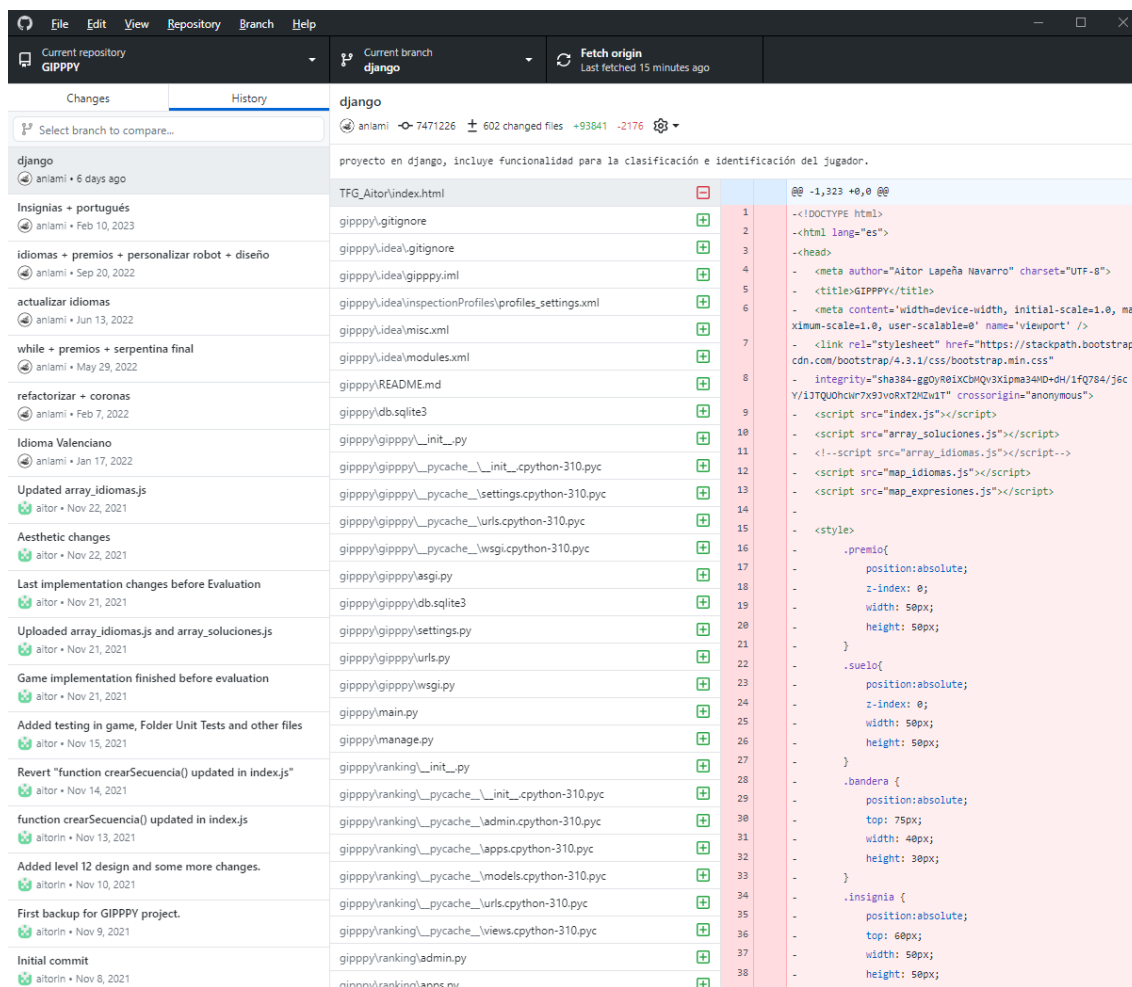


Figura 5.4: Repositorio del código de la aplicación en GitHub

5.1 Ampliaciones

A continuación se detallan las ampliaciones más destacadas que han sido incluidas en el juego.

- **Nuevos idiomas:** se incluye la posibilidad de utilizar el valenciano, alemán, portugués y japonés como idiomas del juego, haciendo que los conceptos lleguen más

²<https://github.com/GIPPPY/>

claramente al alumno según su idioma nativo. Para incluirlos se ha modificado la forma en que se almacenan las cadenas, utilizando un mapa de claves-valor para cada idioma 5.1, en lugar de una colección de cadenas.

```

1  /* Fichero para los maps con los string necesarios para el juego en
   diferentes idiomas.
2  Si se quiere anyadir otro idioma al juego, simplemente crear un nuevo
   map para el idioma.
3  IMPORTANTE respetar los nombres de los indices del map. */
4
5  // ESPANYOL
6  let map_esp = new Map();
7  map_esp.set('avanza', "avanza");
8  map_esp.set('gira', "gira");
9  map_esp.set('titulo', "Juego sencillo para introducir a la programacion
   en Python");
10 ...
11
12 // INGLES
13 let map_eng = new Map();
14 map_eng.set('avanza', "forward");
15 map_eng.set('gira', "turn");
16 map_eng.set('titulo', "Easy game to introduce programming in Python");
17 ...
18
19 // VALENCIANO
20 let map_vlc = new Map();
21 map_vlc.set('avanza', "avanca");
22 map_vlc.set('gira', "gira");
23 map_vlc.set('titulo', "Joc sencill per a introducirse a la programacio
   amb Python");
24 ...
25
26 // PORTUGUES
27 let map_por = new Map();
28 map_por.set('avanza', "forward");
29 map_por.set('gira', "turn");
30 map_por.set('titulo', "Jogo simples para introduzira programacao em
   Python");
31 ...
32
33 // ALEMAN
34 let map_ger = new Map();
35 map_ger.set('avanza', "forward");
36 map_ger.set('gira', "turn");
37 map_ger.set('titulo', "Einfaches Spiel zur Einfuhrung in die Python-
   Programmierung");
38 ...

```

Listing 5.1: Mapeo de idiomas

En la interfaz del juego se incluyen imágenes con las banderas de los nuevos idiomas, invocando la función encargada de cambiar de idioma indicando el idioma a utilizar. Una posible mejora sería utilizar un componente que permita mostrar los idiomas agrupados, como un desplegable, ya que al disponer cada vez de más traducciones ocupan bastante espacio.

```

1 <td>
2   <input type="image" class="bandera" src="bandera_esp.png" style="left
   : 1000px;" onclick="cambiarIdioma('esp')">
3 </td>
4 <td>
5   <input type="image" class="bandera" src="bandera_vlc.png" style="left
   : 1050px;" onclick="cambiarIdioma('vlc')">

```

```

6 </td>
7 <td>
8   <input type="image" class="bandera" src="uk_flag.png" style="left:
9     1100px;" onclick="cambiarIdioma('eng')">
10 </td>
11 <td>
12   <input type="image" class="bandera" src="bandera_portugal.png" style=
13     "left: 1150px;" onclick="cambiarIdioma('por')">
14 </td>
15 <td>
16   <input type="image" class="bandera" src="bandera_alemania.png" style=
17     "left: 1200px;" onclick="cambiarIdioma('ger')">
18 </td>

```

Listing 5.2: Imágenes para los idiomas

- Instrucción "mientras":** en el nivel 10 se introduce la instrucción *mientras*, la cual permite aplicar bucles utilizando la condición *no_choca*. Para conseguirlo se ha revisado el código del juego para permitir el movimiento del robot, incluyendo el mapa para nuevo nivel y las expresiones que permiten utilizarla junto a la cláusula *no_choca*. En 5.3 pueden verse las modificaciones incluidas en la función encargada de validar las líneas de código introducidas por el alumno durante el juego.

```

1 // Funcion que valida cada linea de codigo del panel de codigo.
2 async function comprobarArray(array) {
3   return new Promise(function(resolve, reject) {
4     ....
5     const start = async () => {
6       await asyncForEach(array, async (instruccion) => {
7         await new function() {
8           if (es_bucle) {
9             // instrucciones dentro de la estructura iterativa repite():
10            ....
11           } else if (es_condicional) {
12             // instrucciones dentro de estructura condicional si():
13            ....
14           } else if (es_tc) {
15             // instrucciones dentro de un testcase
16            ....
17           } else if (es_while) { // instruccion MIENTRAS
18             // instrucciones dentro de la estructura iterativa while():
19             if (concuerda_avanza_ident || concuerda_giral_ident ||
20             concuerda_gira2_ident) {
21               instruccion = instruccion.substring(2); // hay que filtrar
22               los espacios (2 espacios antes de la instruccion)
23               array_while.push(instruccion);
24             } else if (concuerda_mientras_no_choca_ident) { // no soporta
25               whiles anidados
26               texto_salida.value = map_idioma.get('errorWhile');
27               resolve(false);
28             } else if (concuerda_repite_ident) { // no soporta bucles
29               anidados
30               texto_salida.value = map_idioma.get('errorWhileBucle');
31               resolve(false);
32             } else if (concuerda_si_norte_ident || concuerda_si_este_ident
33             || concuerda_si_sur_ident || concuerda_si_oeste_ident) { // no
34             soporta combinacion de estructuras
35             texto_salida.value = map_idioma.get('errorWhileCondiciones'
36             );

```

```

30     resolve(false);
31   } else if (concuerta_avanza || concuerta_gira1 ||
concuerta_gira2) { // instrucciones sin espacios, fuera del while
32     es_while = false;
33     crearArrayWhile(array_while, repeticiones).then(() => {
34       array_while = [];
35       array_traducido.push(instruccion);
36     })
37   } else if (concuerta_si_norte || concuerta_si_este ||
concuerta_si_sur || concuerta_si_oeste) { // fin de while, inicio de
condicional
38     es_while = false;
39     crearArrayWhile(array_while, repeticiones).then(() => {
40       array_while = [];
41       es_condicional = true;
42       condicion = instruccion.substring(3);
43     })
44   } else if (concuerta_repite) { // inicio de otro bucle
45     es_while = false;
46     crearArrayWhile(array_while, repeticiones).then(() => {
47       array_while = [];
48       es_bucle = true;
49       repeticiones = instruccion.match(/\d+/);
50     })
51   } else if (concuerta_testcase) { // inicio de un caso de
prueba
52     es_while = false;
53     crearArrayWhile(array_while, repeticiones).then(() => {
54       array_while = [];
55       es_tc = true;
56       array_tcases.push(instruccion);
57     })
58   } else if (concuerta_mientras_no_choca) { // inicio de un
WHILE
59     crearArrayBucle(array_bucle, repeticiones).then(() => {
60       array_bucle = [];
61       es_while = true;
62       array_while.push(instruccion);
63     })
64   } else { // ERROR
65     terminado = depurarError(instruccion);
66     while (!terminado) {
67       terminado = depurarError(instruccion);
68     }
69     resolve(false);
70   }
71   } else {
72     // si la instruccion no esta dentro de bucle ni condicional
ni de un caso de prueba
73     if (concuerta_avanza || concuerta_gira1 || concuerta_gira2) {
74       // instruccion para pushear directamente al array
75       array_traducido.push(instruccion);
76     } else if (concuerta_repite) { // es un bucle
77       es_bucle = true;
78       repeticiones = instruccion.match(/\d+/);
79       ....
80     } else if (concuerta_mientras_no_choca) { // instruccion
MIENTRAS
81       es_while = true;
82       condicion = 'no_choca';
83     } else { // ERROR
84       terminado = depurarError(instruccion);
85       while (!terminado) {
86         terminado = depurarError(instruccion);

```

```

86         }
87         resolve(false);
88     }
89 }
90 });
91
92 if (es_bucle) {
93     crearArrayBucle(array_bucle, repeticiones).then(() => {
94         resolve(true);
95     })
96     ....
97 } else if (es_while) { // instruccion MIENTRAS
98     crearArrayWhile(array_while, condicion).then(() => {
99         resolve(true);
100    })
101 } else {
102     resolve(true);
103 }
104 }
105 start();
106 });
107 }

```

Listing 5.3: Modificaciones para incluir la instrucción 'mientras'

- **Brújula:** se ha incluido la imagen de una brújula para tratar de facilitar el saber la posición del robot, útil a partir del nivel 7.
- **Botón "Parar programa":** a través del botón "Parar programa" se permite detener la ejecución de las instrucciones del robot, iniciando el nivel actual. En los niveles donde se utilicen instrucciones que entren en bucle será práctico ya que permite reiniciar el nivel sin tener que refrescar la pantalla, consiguiendo así que el alumno no pierda el avance de niveles. Se ha modificado el código para habilitar el botón únicamente cuando se inicie el movimiento del robot. Para conseguir que el robot se detenga se ha incluido un mecanismo que permite marcar que van a dejar de ejecutarse más instrucciones, donde al realizar los movimientos del robot se valida si debe detenerse la ejecución, como puede verse en el código listado en 5.4.

```

1 // Recibe un array de instrucciones y lo va recorriendo de manera
  // recursiva. Con condicionales, funciona de otra manera.
2 async function crearSecuencia(instrucciones) {
3     if (pararEjecucion) {
4         cargar_nivel(nivel_actual, false, true);
5         return false;
6     }
7     ....
8 }
9
10 // Funcion para mover al robot un numero de casillas indicado por
   // parametro. Funciona tanto horizontal como verticalmente.
11 function avanza(npasos) {
12     return new Promise(function(resolve, reject) {
13         if (pararEjecucion) {
14             cargar_nivel(nivel_actual, false, true);
15             clearInterval(id);
16             resolve(false);
17             return;
18         }
19         ....
20     }
21 }
22 // Funcion que provoca que el robot gire mediante cambios de imagen.

```

```

23 function gira(ngiros) {
24   return new Promise(function(resolve, reject) {
25     if (pararEjecucion) {
26       cargar_nivel(nivel_actual, false, true);
27       clearInterval(id);
28       resolve(false);
29       return;
30     }
31     ....
32 }

```

Listing 5.4: Modificaciones para detener la ejecución

5.2 Elementos de gamificación

Seguimos con las modificaciones realizadas para incluir elementos de gamificación:

- Barra de navegación:** a través de la barra de navegación el alumno puede acceder rápidamente a los diferentes niveles del juego, puede ver en qué nivel está actualmente, y también permite conocer rápidamente los niveles que ha superado, lo cual puede ser visto también como un elemento de puntuación. En 5.5 puede verse como se incluyen las coronas en la interfaz, haciendo uso de las clases CSS definidas, y de la función *cargar_nivel* encargada de conformar el mapa ubicando los obstáculos. Los identificadores de cada corona son utilizados para ir actualizando el color del texto y la imagen de la corona en caso de encontrar la solución de cada nivel, consiguiendo así ver el avance del alumno y el nivel activo actualmente.

```

1 <td class="img-nivel">
2   
3   <h3 class="txt-nivel" id='txt-nivel1' onclick="cargar_nivel(1,false,
   false)">1</h3>
4 </td>
5 ...
6 <td class="img-nivel">
7   
8   <h3 class="txt-nivel2" id='txt-nivel13' onclick="cargar_nivel(13,
   false,false)">13</h3>
9 </td>

```

Listing 5.5: HTML de las coronas

- Premios:** Se incluyen premios al ir superando niveles, los cuales permiten personalizar el robot. En 5.6 puede verse la definición de las imágenes en el HTML. Cuando el alumno supera un nivel, se gestiona si el total de niveles superados coincide con el número asociado al premio, en cuyo caso se modifica la imagen del premio para utilizar la coloreada en lugar de la gris, permitiendo así que pueda utilizarlo para que el robot lo lleve puesto a través de la función *seleccionar_premio*, pasando por parámetro el premio seleccionado. Se han separado los premios por la ubicación en el robot, los de la cabeza son las gafas, el gorro de fiesta y el birrete, los del cuerpo son el lazo y la medalla, y los de la zona inferior las esposas y el coche. Para conseguirlo se han creado imágenes con todas las combinaciones de premios, mostrando la correspondiente imagen según el premio seleccionado.

```

1 

```

```

2 
3 
4 
5 
6 
7 

```

Listing 5.6: HTML de la barra de premios

Para ajustar los objetos que puede llevar el robot se hace uso de la función **seleccionar_premio** 5.7, la cual separa los objetos que puede llevar según si aparecen en la cabeza, cuerpo o en los pies, permitiendo llevar un máximo de 3 objetos a la vez. Existe una imagen por cada combinación posible, la cual es cargada según la combinación seleccionada por el alumno.

```

1 function seleccionar_premio(nombre) {
2     var obj = document.getElementById(nombre);
3     var obj_src = obj.getAttribute("src");
4     if (obj_src != "/static/img_robot/"+nombre+"K0.png")
5     {
6         if(nombre=="gorrofiesta" || nombre=="gorro" || nombre=="lazo")
7             obj_cabeza = obj_cabeza=="_" + nombre ? "" : "_" + nombre;
8         else if(nombre=="esposas" || nombre=="coche" || nombre=="medalla"
9         )
10            obj_pies = obj_pies=="_" + nombre ? "" : "_" + nombre;
11        else
12            obj_cuerpo = obj_cuerpo=="_" + nombre ? "" : "_" + nombre;
13
14        imagen_robot_frente = "/static/img_robot/robot_frente"+obj_cabeza
15        +obj_cuerpo+obj_pies+".gif";
16        imagen_robot_espalda = "/static/img_robot/robot_espalda"+
17        obj_cabeza+obj_cuerpo+obj_pies+".gif";
18        imagen_robot_derecha = "/static/img_robot/robot_lateral"+
19        obj_cabeza+obj_cuerpo+obj_pies+".gif";
20        imagen_robot_izquierda = "/static/img_robot/robot_lateral_atras"+
21        obj_cabeza+obj_cuerpo+obj_pies+".gif";
22        var img_player = document.getElementById('robot');
23        img_player.src = imagen_robot_frente;
24    }
25 }

```

Listing 5.7: Función para el comportamiento al seleccionar un premio

- Insignias:** El estudiante consigue insignias al superar 4, 8 y 12 niveles. Al superar un nivel, si el total de niveles superados coincide con el número asociado a cada insignia, se muestra la insignia que valora el número de niveles superados, con un máximo de 3 insignias. En 5.8 puede verse el HTML utilizado para incluirlas, haciendo uso de la clase CSS insignia que define sus posiciones y tamaños, utilizando los diferentes identificadores para mostrar la imagen de la insignia en caso de superar un nivel y coincidir el número de niveles que el alumno lleva superados con los

asociados a cada insignia según los valores de las constantes *superados_insignia_X*, como puede verse en 5.9.

```

1 
2 
3 

```

Listing 5.8: HTML de las insignias

```

1 let superados_insignia_1 = 4;
2 let superados_insignia_2 = 8;
3 let superados_insignia_3 = 12;
4 function avanza(npasos) {
5     ...
6     // NIVEL COMPLETADO
7     if (img_nivel_src != "/static/img/coronaOK.png") {
8         img_nivel.src = "/static/img/coronaOK.png";
9         txt_nivel.style.color = "green";
10        niveles_superados = calcular_superados();
11
12        if (niveles_superados == superados_insignia_1) {
13            ganar_insignia("insignia1");
14        }
15        if (niveles_superados == superados_insignia_2) {
16            ganar_insignia("insignia2");
17        }
18        if (niveles_superados == superados_insignia_3) {
19            ganar_insignia("insignia3");
20        }
21        ...
22    }
23    function ganar_insignia(nombre){
24        var obj = document.getElementById(nombre);
25        obj.src = "/static/img/"+nombre+".png";
26    }

```

Listing 5.9: Mostrar la insignia según lo niveles superados

- **Premio final:** Una vez superados todos los niveles se informa al alumno a través de una animación divertida de confeti 5.10. Se trata de un premio final sencillo, que trata de hacer ver al estudiante que ha conseguido superar todos los retos planteados.

```

1 function avanza(npasos) {
2     ...
3     if (niveles_superados == num_niveles){
4         document.body.style.backgroundImage = "url('/static/img/confeti.
  gif')";
5         alert(map_idioma.get('nivelesSuperados'));
6     }
7     ...
8 }

```

Listing 5.10: Mostrar el confeti al superar todos los niveles

- **Tabla de clasificación:** A través del botón *clasificación* se permite al alumno ver un listado con los nombres de los participantes que han superado más número de niveles. Para poder identificar al alumno se utiliza el botón *Identificate*, haciendo uso de un formulario donde se solicita el nombre. En 5.11 pueden verse los elementos

HTML necesarios para el funcionamiento de la clasificación. El div *popup* se utiliza para mostrar la clasificación. El div *popupUsuario* es utilizado para mostrar el formulario donde solicitar el nombre del alumno, ocultándolo al cerrar el formulario a través de la función `ocultarFormUsuario()`.

```

1 <div id="popup"></div>
2 <div id="popupUsuario">
3   <form id="formulario">
4     <label for="nombreUsuario" id="lblNombre"><i class="bi bi-person"></i>
5     > Nombre:</label>
6     <input type="text" id="nombreUsuario" name="nombreUsuario" maxlength=
7     "50" autofocus>
8     <button id="closeFormulario" class="btn btn-secondary"
9     style="position:absolute; top:0px; right:0px; padding:4px;
10    margin:2px"
11    onclick="ocultarFormUsuario()">
12    <i class="bi bi-x-square"></i>
13    </button>
14    <button type="submit" class="btn btn-secondary">OK</button>
15  </form>
16 </div>
17
18 <td>
19   <btn id="usuario" class="btn btn-light" type="button" style="position
20   :relative; left:20px; width:140px;" onclick="cambiarUsuario()">
21   <i class="bi bi-person"></i> Player
22 </btn>
23 </td>
24 <td>
25   <h5 id="txtUsuario" style="position:relative; left:30px; width:385px;
26   text-align:left;"></h5>
27 </td>
28 <td>
29   <btn id="clasificacion" class="btn btn-light" type="button" style="
30   position:relative; margin-right:20px; width:150px;" onclick="
31   mostrarClasificacion()">
32   <i class="bi bi-trophy"></i> Ranking
33 </btn>
34 </td>

```

Listing 5.11: HTML con los elementos necesarios para la clasificación

El botón para identificar al usuario y el texto con el saludo están definidos también en el HTML, donde el botón hace uso de la función `cambiarUsuario()` 5.12 para mostrar el formulario.

```

1 function cambiarUsuario(msg, myYes) {
2   document.getElementById("popupUsuario").style.display = "block";
3 }

```

Listing 5.12: Funcion encargada de mostrar el formulario de identificación

Finalmente aparece el botón para mostrar la tabla de clasificación, el cual hace uso de la función `mostrarClasificacion()` 5.13, encargada de invocar al servicio que inserta el nivel superado del alumno, y recupera las posiciones de los 10 mejores para mostrarlas.

```

1 function mostrarClasificacion () {
2   try {
3     ocultarFormUsuario ();
4     let nombreAux = localStorage.getItem ('nombre');
5     if (nombreAux != null && nombreAux != '' && nombreAux != 'null')
6     {

```

```

6     const token = document.getElementsByName('csrfmiddlewaretoken
7     ')[0].value;
8     // Enviar la solicitud POST y esperar a que se complete
9     fetch('/gippy/nuevo/', {
10    method: 'POST',
11    headers: {
12        'Content-Type': 'application/json',
13        'X-CSRFToken': token
14    },
15    body: JSON.stringify({nombre: nombreAux, nivel:
16    niveles_superados})
17    }).then(response => {
18    petitionRanking();
19    }).catch(error => {
20    console.error('Ha ocurrido un error al guardar la
21    clasificacion:', error);
22    petitionRanking();
23    });
24    }
25    } catch (error) {
26    console.error(error);
27    }
28
29    // Hacer visible el popup
30    document.getElementById("popup").style.display = "block";
31
32    }
33
34    function petitionRanking() {
35    // Petición para obtener la clasificación.
36    fetch('/gippy/ranking/')
37    .then(response => response.text())
38    .then(data => {
39    document.getElementById('popup').innerHTML = data;
40    })
41    .catch(error => {
42    console.error('Ha ocurrido un error al consultar el ranking:',
43    error);
44    });
45    }

```

Listing 5.13: Función encargada de insertar el nivel actual y mostrar la clasificación

Para incluir el elemento de clasificación se ha hecho uso del framework web Django, creando un servicio que permita gestionar la información de la clasificación de los alumnos en una base de datos. Para ponerlo en marcha sobre un servidor Linux Ubuntu, en el cual previamente se ha instalado Python y el administrador de paquetes de Python pip, se han seguido los siguientes pasos básicos:

1. **Python y pip:** Django es un framework de Python, por lo que el servidor debe tenerlo instalado. También se instala el administrador de paquetes de Python pip, el cual permite instalar las dependencias necesarias para ejecutar el proyecto.
2. **Instalar Django:** Es necesario tener instalado Django en el servidor, para realizarlo se ha utilizado los comandos *pip install django* y *pip install django-admin*.
3. **Gestor de entornos virtuales:** Es una buena práctica crear un entorno virtual para un proyecto Django, ya que permite disponer de una versión aislada de Python y las dependencias del proyecto, evitando conflictos con otros proyectos que puedan existir en el servidor. A través del *pip install virtualenvwrapper* se instala el paquete que permite gestionar entornos virtuales.

4. **Crear el proyecto:** Para crear un nuevo proyecto se ha utilizado el comando `django-admin startproject gipppy`, el cual crea la estructura de carpetas y archivos necesaria para comenzar a desarrollar la aplicación. Tras realizar este paso se ha incluido el código fuente de GIPPPY, aprovechando el momento para reestructurar la ubicación de los ficheros 5.5, ajustando el código para referenciarlo adecuadamente.

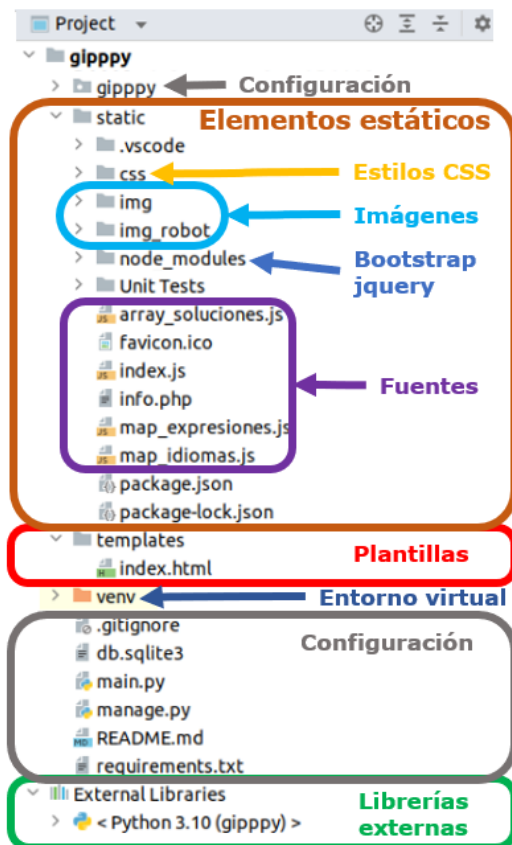


Figura 5.5: Estructura del código fuente

5. **Crear la aplicación:** Para crear la aplicación *ranking* dentro del proyecto se ha utilizado el comando `python manage.py startapp ranking`, creando la estructura de carpetas y archivos necesaria para implementarla.
6. **Definir el modelo de datos:** Un modelo de datos es una representación en código de las tablas de la base de datos que se van a utilizar. Se ha creado una clase en `models.py` dentro de la aplicación con la definición de las tablas, la cual hereda de `models.Model`, donde se han definido los campos y los tipos de datos de la tabla para gestionar la clasificación.
7. **Crear las tablas en la base de datos:** Se han creado las tablas en la base de datos a través de los comandos `python manage.py makemigrations` y `python manage.py migrate`. Se ha optado por utilizar el motor de base de datos SQLite, ya que está incluido en la distribución estándar de Python, es ligero y no requiere de un servidor de base de datos separado.
8. **Crear las vistas:** Una vista es una función o una clase que se encarga de procesar una petición HTTP y devolver una respuesta HTTP. Una vez definido el modelo de datos y creadas las tablas en la base de datos, se han creado las vistas que permitirán interactuar con la base de datos.

9. **Crear las URLs:** Finalmente, se han creado las URL que permitirán acceder a las vistas creadas en el archivo `urls` de la aplicación.

En la imagen 5.6 puede verse la estructura de la aplicación, siguiendo el patrón Modelo-Vista-Controlador o el patrón Modelo-Vista-Plantilla asociada al contenido con el código fuente de la aplicación, destacando la plantilla `ranking.html` utilizada para mostrar el formulario con la clasificación, el modelo `models.py` con la definición de la tabla utilizada para gestionar la información de los niveles que van superando los alumnos, la definición del direccionamiento en `urls.py` asociando las URLs con las vistas, y las vistas con el comportamiento en `views.py`, donde se observa que a través de la ruta `gippyy` se accede a la plantilla `index.html` con la página principal del juego, la ruta `gippyy/ranking` invoca a la vista para consultar los 10 primeros registros de la clasificación y mostrarlos haciendo uso de la plantilla `ranking.html` renderizando los datos de la vista en el HTML y devolverlos al navegador web, y la ruta `gippyy/nuevo` es la encargada de insertar en la base de datos el registro con el nivel superado por el alumno.

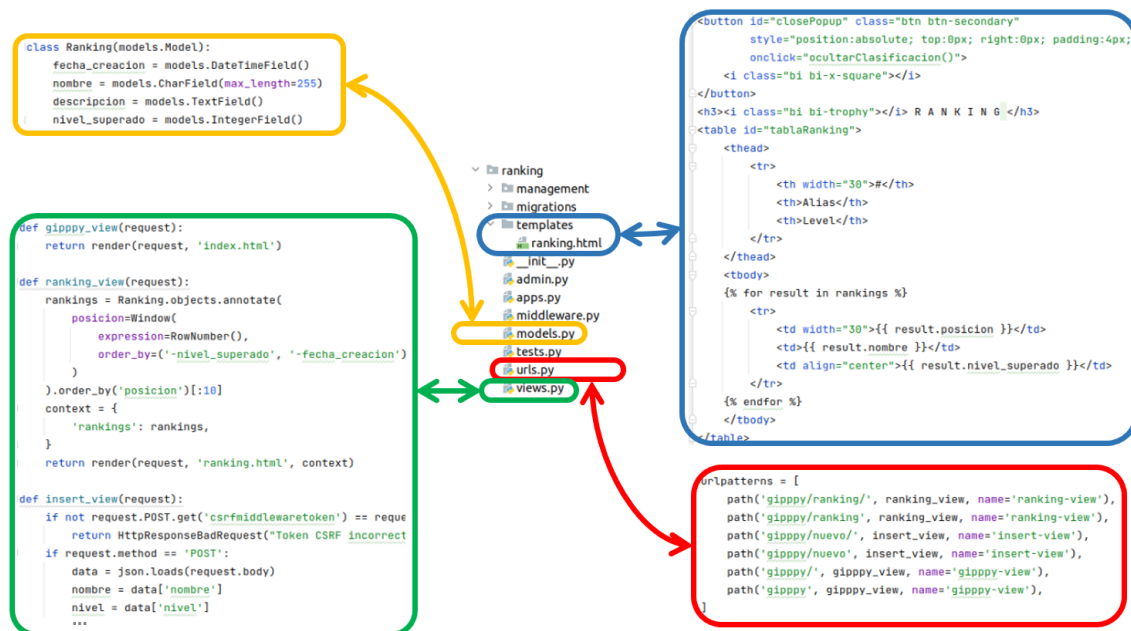


Figura 5.6: Estructura y código de la aplicación Ranking

Desde `gippyy` se consulta la clasificación al pulsar el botón *Clasificación* a través del servicio web definido. Previamente a la consulta, si el alumno se ha identificado se inserta en la base de datos el último nivel que ha superado por, actualizando así la clasificación para que aparezca en caso de estar entre los 10 primeros.

Para evitar un uso malintencionado de la base de datos se plantea utilizar un sistema de verificación por usuario-contraseña, con un correo electrónico para poder recuperar la contraseña, viendo que puede resultar negativo de cara al alumno al requerir de información personal. Para evitarlo se investiga la posibilidad de utilizar cuotas en la base de datos SQLite, y aunque esta opción no aparece incorporada por defecto, sí que puede aplicarse limitando el tamaño de nuestra base de datos a través de una monitorización que realice operaciones de mantenimiento, o utilizando un script que a través de la API `sqlite3` para limitar el tamaño de la base de datos, incluso usar alguna extensión como *SQLite Archiving Command* que permite definir un tamaño máximo para una base de datos que elimina los datos más antiguos al alcanzar el límite establecido. Otra opción es limitar el tamaño del fichero con la base de datos, viendo que como el servidor utiliza una

distribución de Linux Ubuntu puede realizarse el control de cuotas de almacenamiento utilizando el sistema de archivos *Quota*. Finalmente se hace uso de la protección CSRF de Django, quedando el resto de opciones como posibles mejoras.

5.3 Refactorización

Durante la ampliación del juego se realiza una refactorización del código para tratar de hacerlo más legible, claro y fácil de mantener para futuros programadores. A continuación vemos algunos ejemplos de las modificaciones realizadas:

- Las traducciones de los diferentes idiomas estaban en un array, accediendo a cada elemento por la posición. Se pasa a utilizar un mapeo clave/valor [5.14](#) que permite acceder por nombre, simplificando el acceso y consiguiendo un mantenimiento más legible.

```

1 let map_esp = new Map();
2 map_esp.set('avanza', "avanza");
3 map_esp.set('gira', "gira");
4 map_esp.set('titulo', "Juego sencillo para introducir a la programación
   en Python");
5 map_esp.set('btnTraducir', "Traducir a robot");
6 map_esp.set('btnEjecutar', "Ejecutar programa");
7 ...

```

Listing 5.14: Mapeo de idiomas

En [5.15](#) puede verse un ejemplo de uso de las traducciones para el mapeo según el idioma seleccionado.

```

1 function cambiarIdioma(lang) {
2     ...
3     h4.innerHTML = map_idioma.get('titulo');
4     btn_trad.value = map_idioma.get('btnTraducir');
5     btn.value = map_idioma.get('btnEjecutar');
6     btn_parar.value = map_idioma.get('btnParar');

```

Listing 5.15: Ejemplo de uso de traducciones

- Todas las imágenes estaban al mismo nivel, en la raíz del proyecto. Al incluir nuevas imágenes, sobre todo para la personalización del robot a través de premios por superar niveles, se han separado las imágenes del resto de ficheros para tenerlas en las subcarpetas *img* e *img_robot*, con su consecuente modificación en el código fuente de la aplicación para acceder a la nueva ruta.
- Se separan las expresiones regulares que aparecían en el mismo array que contiene las traducciones para tenerlos en un fichero aparte, utilizando también un mapeo clave/valor, como puede verse en [5.16](#).

```

1 /* Fichero para los maps con las expresiones necesarias para el juego en
2 diferentes idiomas. */
3
4 let map_reg_esp = new Map();
5 map_reg_esp.set('avanza',/avanza\\(\\d+\\)/);
6 map_reg_esp.set('avanza_identado',/ avanza\\(\\d+\\)/);
7 map_reg_esp.set('gira_der',/gira\\(\\d+\\)/);
8 map_reg_esp.set('gira_der_identado',/ gira\\(\\d+\\)/);
9 map_reg_esp.set('gira_izq',/gira\\(-\\d+\\)/);
10 map_reg_esp.set('gira_izq_identado',/ gira\\(-\\d+\\)/);
11 map_reg_esp.set('repite',/repite\\(\\d+\\)/);

```

```

12 map_reg_esp.set('repite_identado',/ repite\\(\\d+\\):/);
13 map_reg_esp.set('si_robot_sur',/ si\\(robot_sur\\):/);
14 map_reg_esp.set('si_robot_sur_identado',/ si\\(robot_sur\\):/);
15 map_reg_esp.set('si_robot_norte',/ si\\(robot_norte\\):/);
16 map_reg_esp.set('si_robot_norte_identado',/ si\\(robot_norte\\):/);
17 map_reg_esp.set('si_robot_este',/ si\\(robot_este\\):/);
18 map_reg_esp.set('si_robot_este_identado',/ si\\(robot_este\\):/);
19 map_reg_esp.set('si_robot_oeste',/ si\\(robot_oeste\\):/);
20 map_reg_esp.set('si_robot_oeste_identado',/ si\\(robot_oeste\\):/);
21 map_reg_esp.set('testcase',/ testcase\\d+\\:);
22 map_reg_esp.set('testcase_identado',/ testcase\\d+\\:);
23 map_reg_esp.set('afirma_no_avanza',/ afirma\\(no_avanza\\)/);
24 map_reg_esp.set('afirma_pincha',/ afirma\\(pincha\\)/);
25 map_reg_esp.set('afirma_choca',/ afirma\\(choca\\)/);
26 map_reg_esp.set('afirma_no_cae',/ afirma\\(no_cae\\)/);
27 map_reg_esp.set('mientras_no_choca',/ mientras\\(no_choca\\):/);
28 map_reg_esp.set('mientras_no_choca_identado',/ mientras\\(no_choca\\):/);

```

Listing 5.16: Mapeo de expresiones regulares

- Utilizar más estilos en el html para facilitar la legibilidad y el mantenimiento del código. Para ello se incluye el fichero styles.css, el cual ahora contiene los estilos utilizados en el HTML para los elementos de la página web.

```

1 premio {
2     position: absolute;
3     z-index: 0;
4     width: 50px;
5     height: 50px;
6 }
7 .suelo {
8     position: absolute;
9     z-index: 0;
10    width: 50px;
11    height: 50px;
12 }
13 .bandera {
14    position: absolute;
15    top: 75px;
16    width: 40px;
17    height: 30px;
18 }
19 ...

```

Listing 5.17: Fichero styles.css con los estilos CSS

- Se simplifica la definición de las soluciones del fichero *array_soluciones.js* 5.18 para hacerlas independientes del idioma, ya que hasta ahora se utilizaban por cada solución 2 constantes, 1 para cada idioma, pasando a utilizar una única constante por cada solución, traduciéndolas a través de la función *traducirSolucion* 5.19.

```

1 // Fichero con las soluciones de los niveles NO aleatorios.
2 // Las soluciones estan en formato array para ser comparadas en index.js.
3
4 // NIVEL 1
5 const array_sol_n1 = ["avanza(9)"];
6 // Dejar de mantener las contantes traducidas para cada idioma.
7 //const array_sol_n1_eng = ["forward(9)"];
8
9 //NIVEL 2 y 3 (misma estructura)
10 const array_sol_n2yn3_1 = ["gira(-1)", "avanza(1)", "gira(1)", "avanza(8)",
    "gira(1)", "avanza(1)", "gira(-1)", "avanza(1)"];

```

```

11 const array_sol_n2yn3_2 = ["gira(1)", "avanza(1)", "gira(-1)", "avanza(8)
12    ", "gira(-1)", "avanza(1)", "gira(1)", "avanza(1)"];
...

```

Listing 5.18: Fichero array_soluciones.js

```

1 function traducirSolucion(array_sol) {
2   for (var i=0; i<array_sol.length; i++) {
3     array_sol[i] = array_sol[i].replaceAll('avanza', map_idioma.get('
4     avanza'));
5     array_sol[i] = array_sol[i].replaceAll('gira', map_idioma.get('
6     gira'));
7   }
8   return array_sol;
9 }

```

Listing 5.19: Función traducirSolucion

- Funciones para evitar código repetido

1. **iniciar_nivel 5.20:** Aúna toda la funcionalidad necesaria para iniciar el nivel especificado, diferenciando si se trata de una primera ejecución o se va a parar la ejecución de instrucciones del robot.

```

1 function iniciar_nivel(nivel, prim_ejecucion, esParada) {
2   document.body.style.backgroundImage = "";
3   var btn = document.getElementById('ejecutarprog');
4   var btn_parar = document.getElementById('pararprog');
5   var btn_usuario = document.getElementById('usuario');
6   var texto_py = document.getElementById('tarea');
7   var texto_salida = document.getElementById("output");
8   for(var i=1; i<=num_niveles; i++) {
9     var txt_nivel = document.getElementById('txt-nivel'+i);
10    var img_nivel = document.getElementById('corona-nivel'+i);
11    var img_nivel_src = img_nivel.getAttribute("src");
12    txt_nivel.style.color = img_nivel_src == "coronaOK.png" ? "
13    green" : "rgb(150, 152, 154)";
14  }
15  var txt_niv = document.getElementById('txt-'+nivel);
16  txt_niv.style.color = "darkred";
17  var img_brujula = document.getElementById('brujula');
18  img_brujula.style.height = nivel_actual >= 7 ? '85px' : '0px';
19  img_brujula.style.width = nivel_actual >= 7 ? '85px' : '0px';
20  var continuar;
21  stop = false;
22  btn_parar.disabled = true;
23  if (!prim_ejecucion) {
24    continuar = limpiar_nivel(false);
25    while (!continuar) {
26      continuar = limpiar_nivel(false);
27    }
28  } else {
29    cambiarIdioma("esp");
30    btn.disabled = true;
31    texto_salida.value = "";
32  }
33  if (!esParada)
34    texto_py.value = map_idioma.get(nivel);
35 }

```

Listing 5.20: Función iniciar_nivel

2. **incluir_obstaculo 5.21** e **incluir_obstaculo_aleatorio 5.22**: Estas funciones son muy utilizadas ya que permiten asignar las propiedades de la imagen que genera el mapa del juego.

```

1 function incluir_obstaculo(pos_x, pos_y, tipo) {
2     var img = new Image();
3     img.src = tipo=="obs" ? "obs.gif" : "pinchos.gif";
4     img.style.height = '50px';
5     img.style.width = '50px';
6     img.style.position = 'absolute';
7     img.style.top = 5 + 60*pos_x + 'px';
8     img.style.zIndex = 6;
9     img.style.left = 10 + 60*pos_y + 'px';
10
11     array_obstaculos.push(img);
12     escenario.appendChild(img);
13 }

```

Listing 5.21: incluir_obstaculo

```

1 function incluir_obstaculo_aleatorio(columna, tipo, num_total,
2     num_obs) {
3     var x = Math.floor((Math.random() * num_total) + 1);
4     var i = 0;
5     var j = 0;
6     while (i < num_obs) {
7         // el valor de x decide aleatoriamente donde habra un hueco
8         if (x != j) {
9             incluir_obstaculo(j, columna, tipo);
10            i++;
11        }
12        j++;
13    }

```

Listing 5.22: incluir_obstaculo_aleatorio

Al cargar cada nivel se especifican las propiedades de los objetos que conforman el mapa, permitiendo así facilitar la definición de los niveles, como puede verse en 5.23.

```

1 function cargar_nivel(nivel, prim_ejecucion, esParada) {
2     nivel_actual = nivel;
3     iniciar_nivel('nivel'+nivel, prim_ejecucion, esParada);
4     switch (nivel_actual) {
5         case 1:
6             // NIVEL 1
7             for (var i=0; i<9; i++) {
8                 incluir_obstaculo(2,i, 'obs');
9                 incluir_obstaculo(4,i, 'obs');
10            }
11            break;
12            ...
13            case 11:
14                // NIVEL 11 -> ALEATORIO COMPLETAMENTE
15                incluir_obstaculo_aleatorio(2, 'obs', 7, 6);
16                incluir_obstaculo_aleatorio(4, 'pinchos', 7, 6);
17                incluir_obstaculo_aleatorio(7, 'obs', 7, 6);
18                break;
19            ...
20        }
21    }

```

Listing 5.23: Función cargar_nivel

3. **traducir_solucion 5.24:** Esta función es utilizada para poder reutilizar la definición de las soluciones de los niveles no aleatorios, ya que utiliza el mapeo de idiomas para obtener la secuencia de instrucciones adecuada al idioma utilizado en el juego.

```
1 function traducirSolucion(array_sol) {
2   for (var i=0; i<array_sol.length; i++) {
3     array_sol[i] = array_sol[i].replaceAll('avanza', map_idioma.
4     get('avanza'));
5     array_sol[i] = array_sol[i].replaceAll('gira', map_idioma.get
6     ('gira'));
7   }
8   return array_sol;
9 }
```

Listing 5.24: Función traducirSolucion

4. **validar_solucion 5.25:** Permite encapsular toda la funcionalidad encargada de validar la solución al ejecutar el código indicado por el alumno.

```
1 function validarSolucion(array_sol1, array_sol2) {
2   if (array_sol1 != null && array_sol2 == null) {
3     if (array_traducido.length == array_sol1.length &&
4     array_traducido.every((v,i) => v === array_sol1[i])) {
5       eficiente = 2;
6     } else eficiente = 1;
7   } else if (array_sol1 != null && array_sol2 != null) {
8     if ((array_traducido.length == array_sol1.length ||
9     array_traducido.length == array_sol2.length) &&
10    (array_traducido.every((v,i) => v === array_sol1[i]) ||
11    array_traducido.every((v,i) => v === array_sol2[i]))) {
12     eficiente = 2;
13   } else eficiente = 1;
14 }
```

Listing 5.25: Función validarSolucion

CAPÍTULO 6

Evaluación

Para evaluar la utilidad de aplicar elementos de gamificación al juego se ha realizado una evaluación empírica exploratoria de GIPPPY, con el objetivo de confirmar si motiva a los alumnos a usar el juego y practicar para aprender pensamiento computacional.

El contexto del estudio fue la carrera de ingeniería informática de la Universitat Politècnica de València. La asignatura corresponde a alumnos que actualmente asisten por primera vez al curso de programación. Las clases del curso de programación se han realizado utilizando técnicas tradicionales de enseñanza/aprendizaje como diapositivas de PowerPoint y ejercicios utilizando el editor Python.

En la evaluación empírica, los estudiantes reciben una pequeña descripción de GIPPPY, la url del juego y la instrucción para jugar libremente el tiempo que quieran. Además, instruimos a los alumnos que cuando terminen de jugar, deben responder un cuestionario. El cuestionario consta de preguntas demográficas, preguntas relacionadas con su progreso en GIPPPY, preguntas sobre la percepción del uso de GIPPPY y sobre los elementos de gamificación. Para las preguntas de percepción, seguimos el cuestionario UMUX (Usability Metric for User Experience) [4].

La investigación se realizó en el primer semestre de 2022. Participaron voluntariamente 18 estudiantes. La participación en el experimento estaba relacionada con la puntuación de los estudiantes, además se les instó a que se esforzaran en realizarlo.

6.1 Resultados

En la imagen 6.1 puede verse que el perfil de los participantes corresponde a 10 hombres y 8 mujeres, y que la edad de la mayoría de ellos es de 18 años, variando desde los 17 años hasta los 19 años. Estas son las edades con las que habitualmente acceden los estudiantes a la universidad. Es importante tener en cuenta que el rango de edad de los estudiantes universitarios puede ser muy amplio, desde jóvenes como los que han participado en la encuesta, hasta gente más adulta que decide continuar su educación o cambiar de profesión.

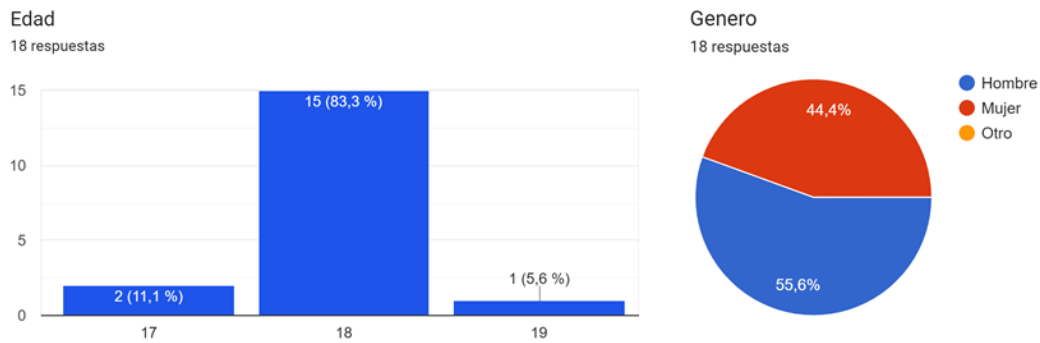


Figura 6.1: Perfil de los participantes

En cuanto a su experiencia en programación 6.2, el 61,1 % de los encuestados respondió que tiene algún conocimiento, pero poco; el 22,2 % de los participantes respondieron que no tenían ningún conocimiento (ya que están iniciando el curso); y el 16,7 % indicó que ya aprendió a programar durante su carrera universitaria. Se puede decir que la mayoría de los encuestados tienen algún conocimiento en programación, aunque en su mayoría es escaso. Además, resulta interesante que un porcentaje significativo de los alumnos ya haya aprendido a programar durante su primer curso universitario.

Respecto al conocimiento sobre Python, el 5,6 % de los participantes respondió si pero no lo he usado; ninguno de los participantes respondieron No, nunca he oído hablar de eso; y el 94,4 % responde Sí, también lo he usado más de una vez. Es gratificante ver que Python es un lenguaje muy conocido entre los encuestados, aún siendo la mayoría alumnos que tienen poco conocimiento en programación. Al disponer la gran mayoría de experiencia previa en Python, al jugar con GIPPPY les permitirá ampliar su conocimiento y habilidades en este lenguaje.

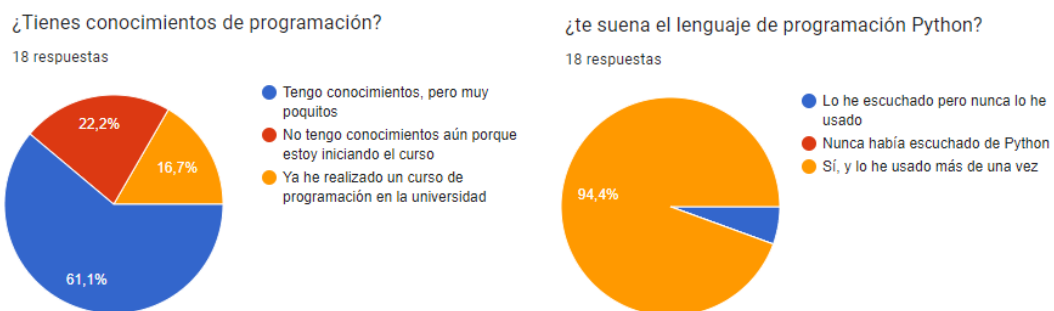


Figura 6.2: Experiencia en programación

En cuanto a la pregunta de si pudieron completar todos los niveles del juego, los resultados indican que el 61,1 % de los participantes pudo hacerlo y el 38,9 % no pudo completar todos los niveles del juego. Ocho de los participantes completaron hasta el nivel 12, y el resto alcanzaron hasta el último nivel, como puede verse en 6.3. La mayoría de los alumnos pudo completar todos los niveles, lo cual es un buen indicador de que es desafiante a la vez que alcanzable. Si se analiza desde otro punto de vista, el que varios de los encuestados se queden en niveles previos debe ser un motivo para mejorar el juego, por ejemplo aportando más elementos de gamificación o una mejor experiencia de aprendizaje.



Figura 6.3: Niveles completados

Los resultados indican que los niveles en los que se divirtieron más 6.4 corresponden al nivel 8 y 10. Esto no es sorprendente ya que en estos niveles los sujetos se involucran en el uso de bucles, condiciones y casos de prueba, que son temas más desafiantes a la par que gratificantes una vez se adquiere su conocimiento, ya que son conceptos más avanzados de programación.

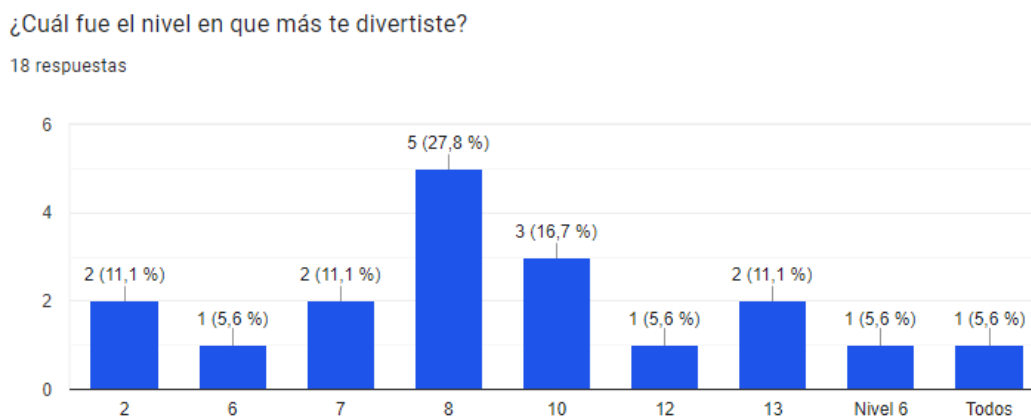


Figura 6.4: Niveles más divertidos

Además, los resultados indican que el nivel más difícil 6.5 es el nivel 13, como era de esperar ya que se pueden usar todas las estructuras y hay aleatoriedad en la posición de los obstáculos, lo cual busca que sea más difícil de completar. Al ser el último nivel hace ver que la dificultad ha ido aumentando con el avance del juego, lo cual puede contribuir a mantener el interés de los alumnos en el aprendizaje.

¿Cuál fue el nivel más complicado?

18 respuestas

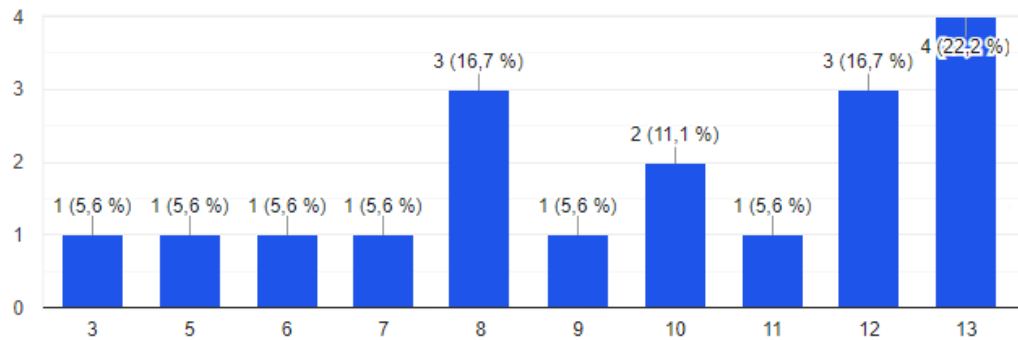


Figura 6.5: Niveles más complicados

Las preguntas del cuestionario UMUX se contestaron mediante una escala Likert de 1: totalmente en desacuerdo a 5: totalmente de acuerdo. Como se puede observar en 6.6, los resultados indican que el 78 % de los sujetos están totalmente de acuerdo o de acuerdo en que GIPPPY les permite aprender a utilizar estructuras básicas en Python, permitiendo decir que se trata de una herramienta útil para el aprendizaje de conceptos en programación. El 67 % de los sujetos está totalmente en desacuerdo o en desacuerdo con que jugar a GIPPPY es una experiencia frustrante, lo cual sugiere que el juego ha sido diseñado de una manera efectiva.



Figura 6.6: Preguntas UMUX

El 94 % de los sujetos está totalmente de acuerdo o de acuerdo en que GIPPPY es fácil de usar, de lo que se puede deducir que el diseño del juego y la interfaz de usuario son intuitivas y están bien aplicadas. Para la última pregunta de UMUX, con respecto a la eficiencia para jugar, hay una tendencia del 66 % de los sujetos que no dedicaron demasiado tiempo a completar los niveles, lo cual aparece reflejado en la imagen 6.7. Esto puede verse como positivo, ya que los alumnos pueden ir ampliando conocimiento al superar niveles sin demasiado esfuerzo, tal vez por estar familiarizados con el lenguaje de programación, o como negativo ya que no han tenido que dedicar mucho tiempo al aprendizaje.



Figura 6.7: Preguntas UMUX

En relación a los elementos del juego que más han gustado, la mayoría ha elegido las recompensas tras superar los niveles, seguido de utilizar niveles, personalizar el robot, y finalmente el premio final y la barra de progreso/estado 6.8, Esto hace ver que son elementos de gamificación que pueden ayudar a motivar a los alumnos a practicar jugando. Ha quedado sin votos la documentación del comportamiento, lo cual puede considerarse como una posible mejora para hacer el juego más atractivo, tal vez permitiendo preguntar al robot y sugiriendo ejemplos en lugar de informarlo todo inicialmente. Se tendrá en cuenta que no se encuentra entre los elementos la tabla de clasificación debido a que cuando se realizó la encuesta no estaba incluida, la cual ya podría incluirse en futuras encuestas.

Contrastando este resultado con el del estudio realizado en el mapeo sistemático se puede deducir que los elementos de gamificación incluidos tras realizar la revisión del estado del arte coinciden con los elegidos por los alumnos como los que más han gustado, haciendo que los alumnos estén más comprometidos e interesados en el aprendizaje a través del juego.



Figura 6.8: Elementos que más han gustado

En la siguiente sección del cuestionario se realizan preguntas abiertas, donde el alumno puede responder en un cuadro de texto. Una pregunta estaba relacionada con el tiempo dedicado a jugar, donde las respuestas van desde los 8 minutos hasta los 40 minutos, siendo la media de 27 minutos, lo cual nos hace ver que es bastante el tiempo que han dedicado a practicar jugando. La siguiente cuestión se centra en si les ha ayudado en el aprendizaje, contestando todos que sí excepto 3 que indican un poco, y 1 que no mucho. Después se les preguntaba sobre si los elementos gamificados les ayudaron a mejorar su motivación, respondiendo que sí excepto 1 no mucho y 1 no. Otra pregunta era

si recomendarían GIPPPY a otros estudiantes para iniciarse en la programación, donde contestaron que sí todos excepto 1 no.

De este grupo de respuestas se deduce que los alumnos dedican un tiempo prudencial a practicar jugando, sintiéndose ayudados por el juego en el aprendizaje, viendo útil es uso de elementos de gamificación para practicar más, incluso recomendando en su gran mayoría el juego para que sea utilizado por otros estudiantes. Destacar que entre los alumnos encuestados aparece una persona que está en desacuerdo con la mayoría de las preguntas, siendo un reto el motivarla a utilizar los juegos para la enseñanza.

La última de las preguntas está relacionada con los elementos de gamificación que añadirían al juego, siendo las respuestas bastante variadas:

- Ni idea
- Sonidos
- if elif else while for...
- Más niveles
- Nada
- Animaciones ganadoras
- Más niveles
- Poder facilitar a las personas mas principiantes mediante respuestas
- Más recompensas
- Skins
- Elementos dinámicos
- Música de fondo mientras completas los niveles, y una música de victoria cada vez que se superan
- Otros robots que tuvieran un patrón de movimiento y con los que si te chocas también pierdes
- Saltar obstáculos
- Ninguno
- La posibilidad de poder tener soluciones hechas para gente principiante
- Ninguno lo veo bien cómo esta

De estas respuestas se podrían destacar elementos de gamificación como animaciones o sonidos al superar los niveles, skins/temas para personalizar la apariencia del juego, ayuda ante problemas en encontrar la solución, más recompensas y más niveles. También se proponen nuevas funcionalidades, como incluir más instrucciones, un comportamiento diferente ante los obstáculos, y otro tipo de robots/obstáculos que sigan un patrón de movimiento. Estas ideas se reciben muy positivamente, valorando el interés en identificar elementos que consideran alentadores para que otros estudiantes busquen mejorar sus habilidades en programación de una manera más entretenida y motivadora, considerándose como posibles trabajos futuros en el desarrollo del juego.

Podemos concluir que los participantes encontraron beneficioso el uso de GIPPY y consideran que el juego es fácil de usar, se divertieron y podría ser un buen complemento para las técnicas ya utilizadas en los cursos de programación para aprender y practicar el pensamiento computacional, sugiriendo que el diseño del juego es efectivo y puede ser útil para estudiantes que busquen mejorar sus habilidades en programación de una manera más entretenida y motivadora.

CAPÍTULO 7

Conclusiones y trabajo futuro

El principal objetivo de este trabajo es ayudar a que los estudiantes aprendan a programar y obtengan el pensamiento computacional a través de la gamificación, ya que se trata de una técnica contrastada que ayuda en la enseñanza, mejorando el rendimiento durante la fase de aprendizaje, utilizándola principalmente para motivar al uso del juego durante más tiempo.

En búsqueda de nuevos elementos de gamificación a aplicar en el juego se ha comprobado que en la literatura se nombran numerosos elementos de gamificación, por lo que se ha realizado una revisión del estado del arte a través de un mapeo sistemático, definiendo la pregunta de investigación **RQ**: *What elements of gamification are used in serious games for programming?* Según los resultados obtenidos se consigue definir una lista de los elementos que más veces aparecen en los artículos analizados, los cuales enumeramos a continuación: tabla de clasificación, uso de niveles, puntuación, insignias, barra de progreso, trofeos o recompensas, avatares, grupos o multijugador, comentarios, y misiones o desafíos.

Seguidamente se han analizado estos elementos para ver cuales serían más adecuados para mejorar la experiencia en el juego, buscando la forma de incluirlos en el juego. Por ejemplo para hacer uso de recompensas se ha optado por premiar al ir superando niveles, permitiendo personalizar la apariencia del personaje principal del juego. Para incluir la barra de progreso se utiliza la imagen de una corona con el número del nivel que representa, apareciendo inicialmente todas en gris, pasando a ser dorada al superar el nivel correspondiente a la corona, lo cual también puede ser visto como un elemento de puntuación al visualizar el total de niveles que va superando. Otro elemento es la tabla de clasificación, para el cual se ha incluido un botón que muestra los diez primeros clasificados según los niveles superados. Las insignias son imágenes de condecoraciones que se muestran en la zona superior del juego. Para los premios obtenidos al ir superando niveles se utiliza la barra lateral con las imágenes de los objetos En la imagen 7.1 puede verse la apariencia de estos elementos en el juego.



Figura 7.1: Principales elementos incluidos en el juego

También se han añadido funcionalidades como la instrucción *mientras*, traducciones a más idiomas, permitir parar la ejecución de instrucciones, y se ha revisado el código fuente de la aplicación para tratar de estructurarlo, haciéndolo más legible y fácil de mantener a través de una refactorización, sin modificar su comportamiento.

A través de la evaluación empírica realizada con la encuesta, se consigue confirmar que el uso de elementos de gamificación motiva a los alumnos a jugar más, consiguiendo que amplíen sus conocimientos en programación y practiquen el pensamiento computacional de una manera lúdica. A través de las respuestas se identifican elementos de gamificación que pueden ser considerados para trabajos futuros, como animaciones con sonidos al obtener los premios por superar los niveles (en lugar de mostrar un mensaje informando del premio, que aparezca una animación con el objeto y un sonido asociado a ganar), permitir seleccionar entre diferentes temas para personalizar el mapa del juego (star wars, videojuegos, anime, minecraft), una revisión de la narrativa incluyendo una posible ayuda cuando el estudiante no pueda encontrar la solución.

Respecto a las limitaciones al continuar con el desarrollo del juego, destacar que en la versión inicial solo se permite un nivel de anidación en las estructuras, y no permite combinarlas (una instrucción *repite* no puede estar dentro de una *mientras*). Para ser más permisivos y admitir varios niveles de anidación habría sido necesario realizar modificaciones profundas en el compilador. Otra limitación es que únicamente aplica la sintaxis y reglas de escritura utilizadas en el lenguaje de programación Python, y añadir nuevos lenguajes conllevaría un cambio importante en el código fuente del juego, lo cual que requeriría de más tiempo para su desarrollo. Un trabajo futuro en este sentido sería rehacer el compilador para permitir nuevas estructuras, instrucciones anidadas, lo cual conllevaría añadir más niveles y recompensas, incluso llegar a permitir seleccionar entre diferentes sintaxis asociadas a reglas y estructuras que se utilicen en otros lenguajes de programación, ampliando así los conocimientos a adquirir por el alumno.

Si bien los resultados de la evaluación empírica realizada son alentadores y sugieren que el juego es una herramienta útil para que los estudiantes aprendan programación de manera más lúdica y motivadora, se hubiese deseado haber podido realizarla en más cursos de programación, disponiendo así de aún más información sobre la utilidad de GIPPPY como herramienta de aprendizaje.

Una vía de mejora del juego sería aplicar un diseño responsivo que se adapte a diferentes tamaños, incluso realizar una revisión completa para hacer que GIPPPY sea más accesible, ya que como puede verse en la imagen 7.2 al realizar un análisis de accesibilidad a través de herramientas web^{1 2} aparecen problemas, principalmente en la percepción.



Figura 7.2: Análisis de accesibilidad del juego

Puede continuarse con el desarrollo del juego, por ejemplo realizando una revisión con más elementos gamificación, o aportando más opciones para la tabla de clasificación, como tener la posibilidad de definir el curso donde participa el alumno para ver el avance comparado con el resto de compañeros, o utilizar más tipos de obstáculos que dificulten la solución en niveles avanzados, incluir un nivel final donde la solución sea genérica y permita resolver cualquier nivel.

¹<https://wave.webaim.org/>

²<https://www.tawdis.net/>

Bibliografía

- [1] C Abt. *Seerous Games*. University Press of America, 2002.
- [2] Andreas Beelmann, M. Petticrew y H. Roberts. «Systematic reviews in the social sciences. A practical guide: (2005). London: Blackwell. ISBN13: 978-1-4051-2110-1 (hardcover)/ISBN10: 1-4051-2110-6 (hardcover); ISBN13: 978-1-4051-2111-8 (pbk)/ISBN10: 1-4051-2111-4 (pbk)». En: *European Psychologist - EUR PSYCHOL* 11 (ene. de 2006), págs. 244-245. DOI: [10.1027/1016-9040.11.3.244](https://doi.org/10.1027/1016-9040.11.3.244).
- [3] Jeannette M. Wing. «Computational Thinking». En: *Commun. ACM* 49.3 (mar. de 2006), págs. 33-35. ISSN: 0001-0782. DOI: [10.1145/1118178.1118215](https://doi.org/10.1145/1118178.1118215). URL: <https://doi.org/10.1145/1118178.1118215>.
- [4] Kraig Finstad. «The Usability Metric for User Experience». En: *Interacting with Computers* 22 (sep. de 2010), págs. 323-327. DOI: [10.1016/j.intcom.2010.04.004](https://doi.org/10.1016/j.intcom.2010.04.004).
- [5] Thomas M. Connolly y col. «A systematic literature review of empirical evidence on computer games and serious games». En: *Computers & Education* 59.2 (2012), págs. 661-686. ISSN: 0360-1315. DOI: <https://doi.org/10.1016/j.compedu.2012.03.004>. URL: <https://www.sciencedirect.com/science/article/pii/S0360131512000619>.
- [6] Kevin Browne y Christopher Anand. «Gamification and Serious Game Approaches for Introductory Computer Science Tablet Software». En: *Proceedings of the First International Conference on Gameful Design, Research, and Applications*. Gamification '13. Toronto, Ontario, Canada: Association for Computing Machinery, 2013, págs. 50-57. ISBN: 9781450328159. DOI: [10.1145/2583008.2583015](https://doi.org/10.1145/2583008.2583015). URL: <https://doi.org/10.1145/2583008.2583015>.
- [7] Barbara Kitchenham y Pearl Brereton. «A systematic review of systematic review process research in software engineering». En: *Information and Software Technology* 55.12 (2013), págs. 2049-2075. ISSN: 0950-5849. DOI: <https://doi.org/10.1016/j.infsof.2013.07.010>. URL: <https://www.sciencedirect.com/science/article/pii/S0950584913001560>.
- [8] Oriol Borrás Gené, Margarita Martínez Núñez y Ángel Fidalgo Blanco. «Gamification in MOOC: Challenges, Opportunities and Proposals for Advancing MOOC Model». En: *Proceedings of the Second International Conference on Technological Ecosystems for Enhancing Multiculturality*. TEEM '14. Salamanca, Spain: Association for Computing Machinery, 2014, págs. 215-220. ISBN: 9781450328968. DOI: [10.1145/2669711.2669902](https://doi.org/10.1145/2669711.2669902). URL: <https://doi.org/10.1145/2669711.2669902>.
- [9] Afef Ghannem. «Characterization of Serious Games Guided by the Educational Objectives». En: *Proceedings of the Second International Conference on Technological Ecosystems for Enhancing Multiculturality*. TEEM '14. Salamanca, Spain: Association for Computing Machinery, 2014, págs. 227-233. ISBN: 9781450328968. DOI: [10.1145/2669711.2669904](https://doi.org/10.1145/2669711.2669904). URL: <https://doi.org/10.1145/2669711.2669904>.

- [10] Lassi Haaranen y col. «How (Not) to Introduce Badges to Online Exercises». En: *Proceedings of the 45th ACM Technical Symposium on Computer Science Education*. SIGCSE '14. Atlanta, Georgia, USA: Association for Computing Machinery, 2014, págs. 33-38. ISBN: 9781450326056. DOI: [10.1145/2538862.2538921](https://doi.org/10.1145/2538862.2538921). URL: <https://doi.org/10.1145/2538862.2538921>.
- [11] Roel Wieringa. «Design Science Methodology for Information Systems and Software Engineering». En: *Springer Berlin Heidelberg*. 2014.
- [12] Philip Mildner, Nicolas Stamer y Wolfgang Effelsberg. «From Game Characteristics to Effective Learning Games». En: *Serious Games*. Ed. por Stefan Göbel y col. Cham: Springer International Publishing, 2015, págs. 51-62. ISBN: 978-3-319-19126-3.
- [13] Ana Carolina Tomé Klock y col. «Gamification in e-Learning Systems: A Conceptual Model to Engage Students and Its Application in an Adaptive e-Learning System». En: *Learning and Collaboration Technologies*. Ed. por Panayiotis Zaphiris y Andri Ioannou. Cham: Springer International Publishing, 2015, págs. 595-607. ISBN: 978-3-319-20609-7.
- [14] Ana Carolina Tomé Klock y col. «Gamification in e-Learning Systems: A Conceptual Model to Engage Students and Its Application in an Adaptive e-Learning System». En: *Learning and Collaboration Technologies*. Ed. por Panayiotis Zaphiris y Andri Ioannou. Cham: Springer International Publishing, 2015, págs. 595-607. ISBN: 978-3-319-20609-7.
- [15] Titus Barik, Emerson Murphy-Hill y Thomas Zimmermann. «A perspective on blending programming environments and games: Beyond points, badges, and leaderboards». En: *2016 IEEE Symposium on Visual Languages and Human-Centric Computing (VL/HCC)*. 2016, págs. 134-142. DOI: [10.1109/VLHCC.2016.7739676](https://doi.org/10.1109/VLHCC.2016.7739676).
- [16] Elizabeth A. Boyle y col. «An update to the systematic literature review of empirical evidence of the impacts and outcomes of computer games and serious games». En: *Computers & Education* 94 (2016), págs. 178-192. ISSN: 0360-1315. DOI: <https://doi.org/10.1016/j.compedu.2015.11.003>. URL: <https://www.sciencedirect.com/science/article/pii/S0360131515300750>.
- [17] Matthias Budde y col. «Sensified Gaming: Design Patterns and Game Design Elements for Gameful Environmental Sensing». En: *Proceedings of the 13th International Conference on Advances in Computer Entertainment Technology*. ACE '16. Osaka, Japan: Association for Computing Machinery, 2016. ISBN: 9781450347730. DOI: [10.1145/3001773.3001832](https://doi.org/10.1145/3001773.3001832). URL: <https://doi.org/10.1145/3001773.3001832>.
- [18] Alessandra Antonaci y col. «Towards Implementing Gamification in MOOCs». En: *Games and Learning Alliance*. Ed. por João Dias, Pedro A. Santos y Remco C. Veltkamp. Cham: Springer International Publishing, 2017, págs. 115-125. ISBN: 978-3-319-71940-5.
- [19] Ian Arawjo y col. «Teaching Programming with Gamified Semantics». En: *Proceedings of the 2017 CHI Conference on Human Factors in Computing Systems*. CHI '17. Denver, Colorado, USA: Association for Computing Machinery, 2017, págs. 4911-4923. ISBN: 9781450346559. DOI: [10.1145/3025453.3025711](https://doi.org/10.1145/3025453.3025711). URL: <https://doi.org/10.1145/3025453.3025711>.
- [20] Firas Layth Khaleel y col. «Gamification-based learning framework for a programming course». En: *2017 6th International Conference on Electrical Engineering and Informatics (ICEEI)*. Nov. de 2017, págs. 1-6. DOI: [10.1109/ICEEI.2017.8312377](https://doi.org/10.1109/ICEEI.2017.8312377).

- [21] Richard N. Landers, Michael B. Armstrong y Andrew B. Collmus. «How to Use Game Elements to Enhance Learning: Applications of the Theory of Gamified Learning». En: *Serious Games and Edutainment Applications : Volume II*. Ed. por Minhua Ma y Andreas Oikonomou. Cham: Springer International Publishing, 2017, págs. 457-483. ISBN: 978-3-319-51645-5. DOI: [10.1007/978-3-319-51645-5_21](https://doi.org/10.1007/978-3-319-51645-5_21). URL: https://doi.org/10.1007/978-3-319-51645-5_21.
- [22] Elisa D. Mekler y col. «Towards understanding the effects of individual gamification elements on intrinsic motivation and performance». En: *Computers in Human Behavior* 71 (2017), págs. 525-534. ISSN: 0747-5632. DOI: <https://doi.org/10.1016/j.chb.2015.08.048>. URL: <https://www.sciencedirect.com/science/article/pii/S0747563215301229>.
- [23] Mariana Peixoto y Carla Silva. «A Gamification Requirements Catalog for Educational Software: Results from a Systematic Literature Review and a Survey with Experts». En: *Proceedings of the Symposium on Applied Computing*. SAC '17. Marrakech, Morocco: Association for Computing Machinery, 2017, págs. 1108-1113. ISBN: 9781450344869. DOI: [10.1145/3019612.3019752](https://doi.org/10.1145/3019612.3019752). URL: <https://doi.org/10.1145/3019612.3019752>.
- [24] Rob van Roy y Bieke Zaman. «Why Gamification Fails in Education and How to Make It Successful: Introducing Nine Gamification Heuristics Based on Self-Determination Theory». En: *Serious Games and Edutainment Applications : Volume II*. Ed. por Minhua Ma y Andreas Oikonomou. Cham: Springer International Publishing, 2017, págs. 485-509. ISBN: 978-3-319-51645-5. DOI: [10.1007/978-3-319-51645-5_22](https://doi.org/10.1007/978-3-319-51645-5_22). URL: https://doi.org/10.1007/978-3-319-51645-5_22.
- [25] Valerie J. Shute, Chen Sun y Jodi Asbell-Clarke. «Demystifying computational thinking». En: *Educational Research Review* 22 (2017), págs. 142-158.
- [26] Mauricio R A Souza y col. «Games for Learning: Bridging Game-Related Education Methods to Software Engineering Knowledge Areas». En: *Proceedings of the 39th International Conference on Software Engineering: Software Engineering and Education Track*. ICSE-SEET '17. Buenos Aires, Argentina: IEEE Press, 2017, págs. 170-179. ISBN: 9781538626719. DOI: [10.1109/ICSE-SEET.2017.17](https://doi.org/10.1109/ICSE-SEET.2017.17). URL: <https://doi.org/10.1109/ICSE-SEET.2017.17>.
- [27] Jenilyn L. Agapito y Ma. Mercedes T. Rodrigo. «Investigating the Impact of a Meaningful Gamification-Based Intervention on Novice Programmers' Achievement». En: *Artificial Intelligence in Education*. Ed. por Carolyn Penstein Rosé y col. Cham: Springer International Publishing, 2018, págs. 3-16. ISBN: 978-3-319-93843-1.
- [28] Andreas Alexiou y Michaéla C. Schippers. «Digital game elements, user experience and learning: A conceptual framework». En: *Education and Information Technologies*. Springer International Publishing, 2018.
- [29] Francisco J. Gallego-Durán y Carlos J. Villagrà Arnedo. «Enchanted Talk: Multiplayer Gamification Using Google Spreadsheets». En: *Proceedings of the Sixth International Conference on Technological Ecosystems for Enhancing Multiculturality*. TEEM'18. Salamanca, Spain: Association for Computing Machinery, 2018, págs. 668-673. ISBN: 9781450365185. DOI: [10.1145/3284179.3284290](https://doi.org/10.1145/3284179.3284290). URL: <https://doi.org/10.1145/3284179.3284290>.
- [30] Gabriela Martins de Jesus y col. «Gamification in Software Testing: A Characterization Study». En: *Proceedings of the III Brazilian Symposium on Systematic and Automated Software Testing*. SAST '18. SAO CARLOS, Brazil: Association for Computing Machinery, 2018, págs. 39-48. ISBN: 9781450365550. DOI: [10.1145/3266003.3266007](https://doi.org/10.1145/3266003.3266007). URL: <https://doi.org/10.1145/3266003.3266007>.

- [31] Mohsen M. Jozani, Michele Maasberg y Emmanuel Ayaburi. «Slayers vs Slackers: An Examination of Users' Competitive Differences in Gamified IT Platforms Based on Hedonic Motivation System Model». En: *Learning and Collaboration Technologies. Learning and Teaching*. Ed. por Panayiotis Zaphiris y Andri Ioannou. Cham: Springer International Publishing, 2018, págs. 164-172. ISBN: 978-3-319-91152-6.
- [32] B. Marín y col. «An Empirical Investigation on the Benefits of Gamification in Programming Courses». En: *ACM Trans. Comput. Educ.* 19.1 (nov. de 2018). DOI: [10.1145/3231709](https://doi.org/10.1145/3231709). URL: <https://doi.org/10.1145/3231709>.
- [33] Frieska Angelia y Suharjito. «Improving English Learning through Game Using 6–11 MDA Framework». En: *2019 12th International Conference on Information Communication Technology and System (ICTS)*. Jun. de 2019, págs. 21-26. DOI: [10.1109/ICTS.2019.8850951](https://doi.org/10.1109/ICTS.2019.8850951).
- [34] Darina Dicheva, Keith Irwin y Christo Dichev. «OneUp: Engaging Students in a Gamified Data Structures Course». En: *Proceedings of the 50th ACM Technical Symposium on Computer Science Education*. SIGCSE '19. Minneapolis, MN, USA: Association for Computing Machinery, 2019, págs. 386-392. ISBN: 9781450358903. DOI: [10.1145/3287324.3287480](https://doi.org/10.1145/3287324.3287480). URL: <https://doi.org/10.1145/3287324.3287480>.
- [35] Hadi Hosseini y Laurel Perweiler. «Are You Game?» En: *Proceedings of the 50th ACM Technical Symposium on Computer Science Education*. SIGCSE '19. Minneapolis, MN, USA: Association for Computing Machinery, 2019, págs. 866-872. ISBN: 9781450358903. DOI: [10.1145/3287324.3287411](https://doi.org/10.1145/3287324.3287411). URL: <https://doi.org/10.1145/3287324.3287411>.
- [36] Gabriela Martins de Jesus y col. «Is It Worth Using Gamification on Software Testing Education? An Experience Report». En: *Proceedings of the XVIII Brazilian Symposium on Software Quality*. SBQS'19. Fortaleza, Brazil: Association for Computing Machinery, 2019, págs. 178-187. ISBN: 9781450372824. DOI: [10.1145/3364641.3364661](https://doi.org/10.1145/3364641.3364661). URL: <https://doi.org/10.1145/3364641.3364661>.
- [37] Firas Layth Khaleel, Noraidah Sahari Ashaari y Tengku Siti Meriam Tengku Wook. «Designing Usable Gamified Programming Learning Website». En: *2019 International Conference on Electrical Engineering and Informatics (ICEEI)*. Jul. de 2019, págs. 164-169. DOI: [10.1109/ICEEI47359.2019.8988818](https://doi.org/10.1109/ICEEI47359.2019.8988818).
- [38] Adriano Lages dos Santos y col. «A Systematic Mapping Study on Game Elements and Serious Games for Learning Programming». En: *Computer Supported Education*. Ed. por Bruce M. McLaren y col. Cham: Springer International Publishing, 2019, págs. 328-356. ISBN: 978-3-030-21151-6.
- [39] P. Vranešić, K. Aleksić-Maslač y B. Sinković. «Influence of Gamification Reward System on Student Motivation». En: *2019 42nd International Convention on Information and Communication Technology, Electronics and Microelectronics (MIPRO)*. Mayo de 2019, págs. 766-772. DOI: [10.23919/MIPRO.2019.8756848](https://doi.org/10.23919/MIPRO.2019.8756848).
- [40] Hani Dewi Ariessanti y col. «Systematic Literature Review Strategy Gamification Through Snake Ladder Game». En: *2020 International Conference on Informatics, Multimedia, Cyber and Information System (ICIMCIS)*. Nov. de 2020, págs. 1-7. DOI: [10.1109/ICIMCIS51567.2020.9354285](https://doi.org/10.1109/ICIMCIS51567.2020.9354285).
- [41] Yen Air Caballero González, Ana García-Valcárcel Muñoz-Repiso y col. «Fortaleciendo el pensamiento computacional y habilidades sociales mediante actividades de aprendizaje con robótica educativa en niveles escolares iniciales». En: *Pixel-Bit* (2020).

- [42] Jose Figueiredo y Francisco Jose Garcia-Peñalvo. «Increasing student motivation in computer programming with gamification». En: *2020 IEEE Global Engineering Education Conference (EDUCON)*. Abr. de 2020, págs. 997-1000. DOI: [10.1109/EDUCON45650.2020.9125283](https://doi.org/10.1109/EDUCON45650.2020.9125283).
- [43] Lennon Sales Furtado y Sandro Ronaldo Bezerra Oliveira. «A teaching proposal for the software measurement process using gamification: an experimental study». En: *2020 IEEE Frontiers in Education Conference (FIE)*. Oct. de 2020, págs. 1-8. DOI: [10.1109/FIE44824.2020.9274194](https://doi.org/10.1109/FIE44824.2020.9274194).
- [44] Natalia Limantara y col. «Mechanics, Dynamics, and Aesthetics Framework on Gamification at University». En: *2020 International Conference on Informatics, Multimedia, Cyber and Information System (ICIMCIS)*. Nov. de 2020, págs. 34-39. DOI: [10.1109/ICIMCIS51567.2020.9354271](https://doi.org/10.1109/ICIMCIS51567.2020.9354271).
- [45] Joel Prieto Andreu. «Una revisión sistemática sobre gamificación, motivación y aprendizaje en universitarios». En: *Teoría de la Educación. Revista Interuniversitaria* 32 (ene. de 2020), pág. 73. DOI: [10.14201/teri.20625](https://doi.org/10.14201/teri.20625).
- [46] Ian T. Dykens y col. «Towards a Unified Model of Gamification and Motivation». En: *Adaptive Instructional Systems. Design and Evaluation*. Ed. por Robert A. Sottolare y Jessica Schwarz. Cham: Springer International Publishing, 2021, págs. 53-70. ISBN: 978-3-030-77857-6.
- [47] Aitor Lapeña Navarro. «Diseño y desarrollo de un juego de navegador para comenzar a aprender construcciones básicas de programación». En: (2021). URL: <https://riunet.upv.es/handle/10251/179241>.
- [48] Miguel Ehécatl Morales-Trujillo y Gabriel Alberto García-Mireles. «Gamification and SQL: An Empirical Study on Student Performance in a Database Course». En: *ACM Trans. Comput. Educ.* 21.1 (dic. de 2021). DOI: [10.1145/3427597](https://doi.org/10.1145/3427597). URL: <https://doi.org/10.1145/3427597>.
- [49] Ryan Macdonell Andrias, Mohd Shahrizal Sunar y Stephen Laison Sondoh. «Adaptive Gamification: User/Player Type and Game Elements Mapping». En: *Intelligent Technologies for Interactive Entertainment*. Ed. por Zhihan Lv y Houbing Song. Cham: Springer International Publishing, 2022, págs. 242-256. ISBN: 978-3-030-99188-3.
- [50] Ryan Macdonell Andrias, Mohd Shahrizal Sunar y Stephen Laison Sondoh. «Adaptive Gamification: User/Player Type and Game Elements Mapping». En: *Intelligent Technologies for Interactive Entertainment*. Ed. por Zhihan Lv y Houbing Song. Cham: Springer International Publishing, 2022, págs. 242-256. ISBN: 978-3-030-99188-3.
- [51] Rakel Gamito y col. «El desarrollo del pensamiento computacional en educación: valoración basada en una experiencia con Scratch». En: *Innoeduca. International Journal of Technology and Educational Innovation* 8.1 (2022), págs. 59-74.
- [52] Coolio-Niato. *Lightbot*. Último acceso: 3 de noviembre de 2022. URL: <https://lightbot.com/>.
- [53] David Guerrero López. *Introducción a la programación*. Último acceso: 3 de noviembre de 2022. URL: <http://personales.upv.es/daguelo/INF/robot/>.
- [54] *La Academia de la Tortuga*. Último acceso: 3 de noviembre de 2022. URL: <https://turtleacademy.com/>.
- [55] Toxicode. *Compute It*. Último acceso: 3 de noviembre de 2022. URL: <https://compute-it.toxicode.fr/>.